1. TOKENIZATION

1.Word piece

2.byte pair encoding

3.unigramLM

Encoding positions:

Transformers processes input sequences in parallel and independently of each other

1. Alibi : It subtracts a scalar bias from the attention score that increases with the distance between token positions
2. Rope : It rotates query and key representations at an angle proportional to the token absolute position

Attention in LLMS:

The attention assigns weights to input tokens based on importance of the model

1. **Self Attention**
2. **Cross Attention**
3. **Sparse Attention**
4. **Flash Attention**

Activation functions:

1. **RELU**
2. **GELU**
3. **GLU variants**

Distributed LLM Training

1. **Data Parallelsim : It replicates model on multiple devices**
2. **Tensor parallesim : Tensor computation across devices**
3. **pipeline parallesim : It shards model layers across different devices**
4. **Model parallesim : Combination of tensor and pipeline**
5. **3D Parallesim : combination of data , tensor and model parallesim**
6. **Optimizer Parallesim : It implements optimiser state across devices to reduce memory consumption**

Libraries

1. **Transformers**
2. **Deep speed**
3. **megatron-LM**
4. **JAX**
5. **colossal AI**
6. **BM trian**
7. FastMoe

Frameworks

1. Mindspore
2. pytorch
3. Tensorflow
4. Mxnet

Data Preprocessing

1. Classifier based : This approach train a classifier on high quality data and predict the quality of text for filtering
2. Heuristics based: The employ some rules for filtering like language metrics, statistics and keywords

Data Deduplication : Data Deduplication is one of the preprocessing steps can be performed at multiple levels like sentences and documents

Architectures

1. Encoder Decoder: The architecture processes inputs through the encoder and passes the intermediate representation to the decoder to generate the output.
2. Causal Decoder : The architecture that does not have encoder and generates output using decoder
3. prefix Decoder: The attention calculation is not strictly dependant on the past information and the attention is bidirectional.

Pre-Training objectives

1. Full Language Modelling
2. Prefix LanguageModelling
3. Masked Language Modelling
4. Unified Language Modelling

Fine tuning

1. Transfer learning : To improve the performance for a downstream task pertained models are fine tuned with specified task
2. Instruction tuning : Instructions generally comprise multi-task data in plain nlp guiding the model to respond the prompt And input
3. Alignment tuning : Alignment involves asking alms to generate unexpected responses and then updating their parameters.

Pretrained LLMs

1. T5 : Encoder and decoder with shared parameters perform equivalently when parameters are not shared.

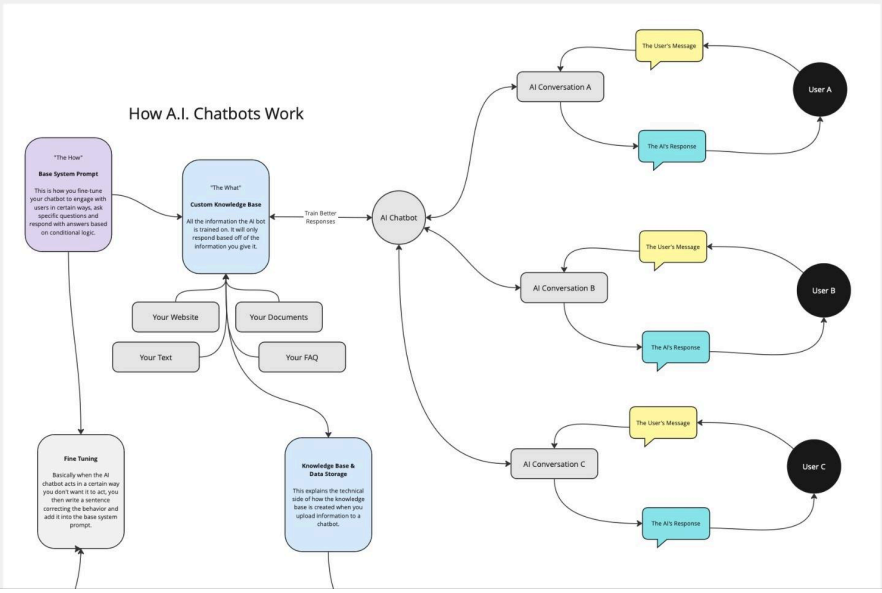
Fine tuning model layers work better than conventional way of training on only classification layers.

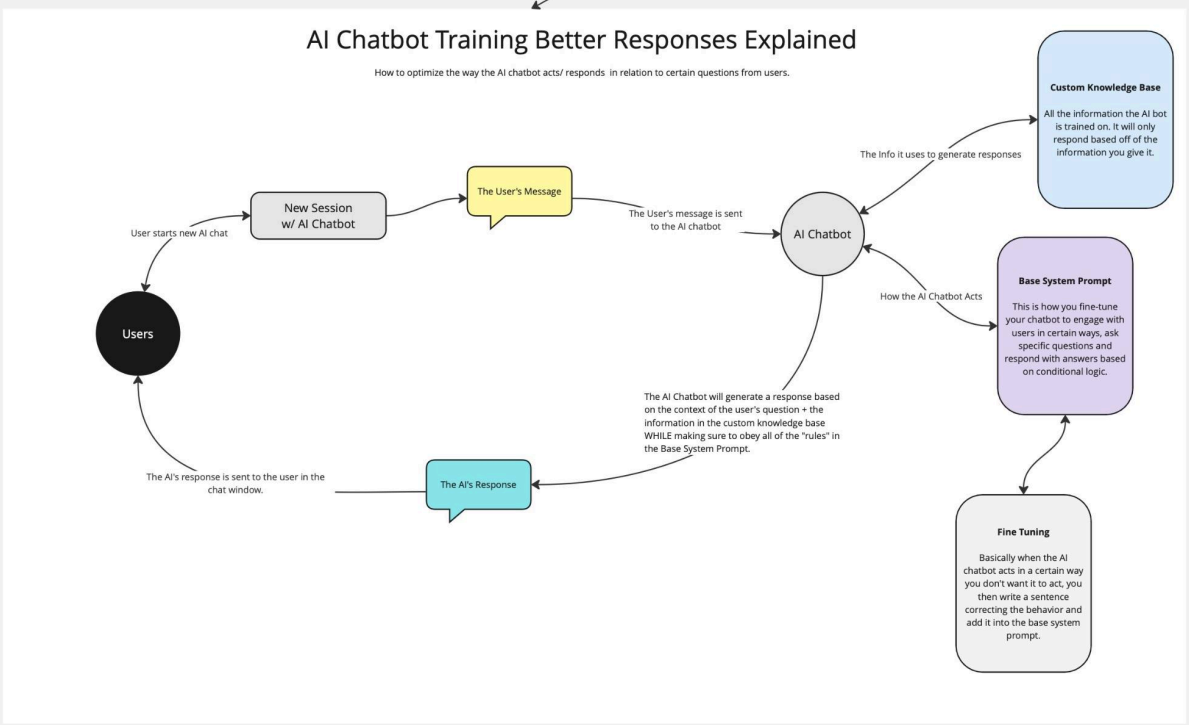
1. GPT-3: Few shot performance of llms better than the zero shot suggesting that alms are meta learners.
2. mT5 : Large multi-lingual models perform equivalently to single language models on down stream tasks however, smaller multilingual models perform worse.
3. PanGu-Alpha : LLMS are good at a few shot capabilities.
4. CPM-2: Prompt fine tuning takes more time to converge as compared to full model fine tuning.

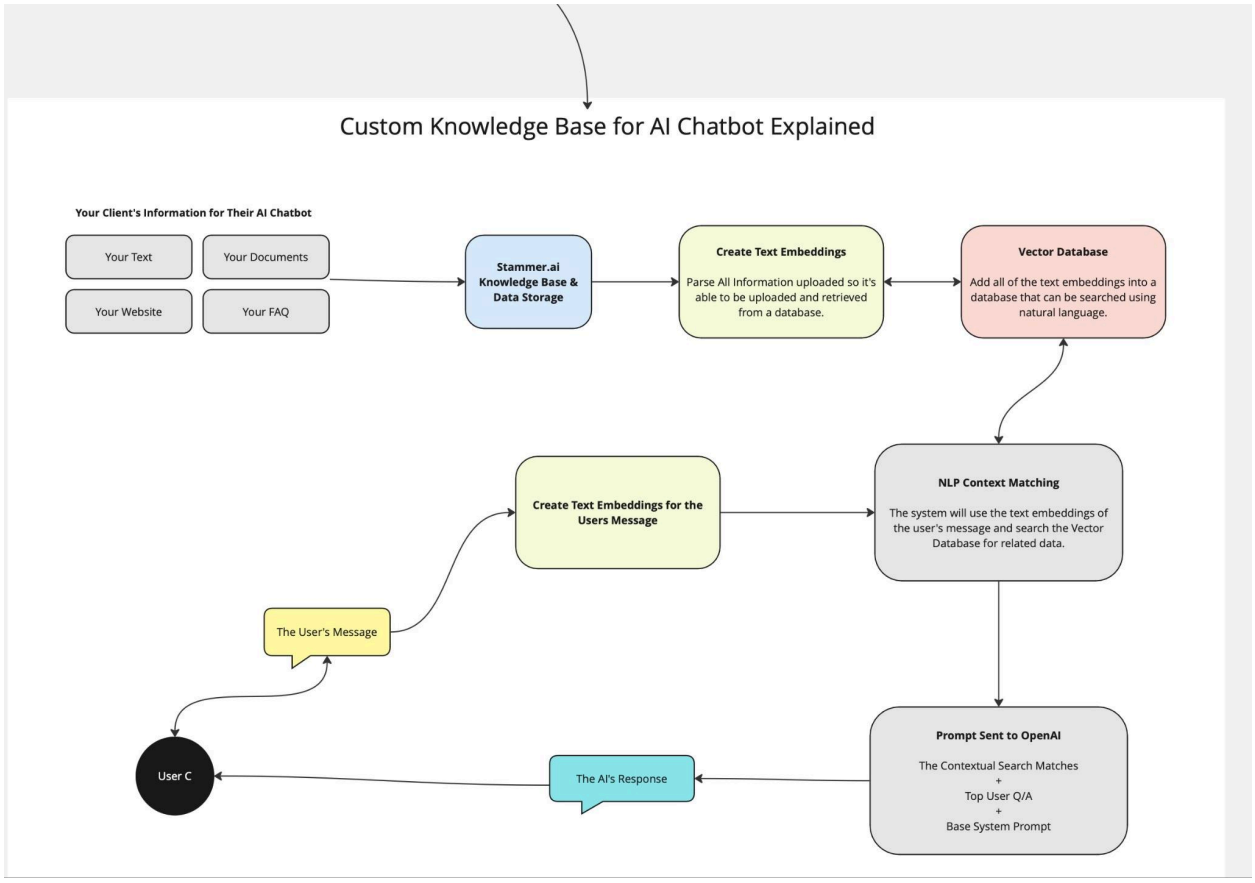
Inserting prompt tokens in between sentences can allow the model to understand relations between sentences and long sequences.

1. codex: This LLM focuses on code evaluations and introduces a novel way of selecting the best code examples.
2. ERNIE 3.0: ERNIE 3.0 shows that a modular Llm architecture with a universal representation module and task specific representation module helps in fine tuning phase.

AI CHAT BOT WORKING PRINCIPAL:


How A.I. Chatbots Work
"The How"
Base System Prompt
This is how you fine-tune your chatbot to engage with users in certain ways, ask specific questions and respond with answers based on conditional logic.
"The What"
Custom Knowledge Base
All the information the AI bot is trained on. It will only respond based off of the information you give it.
Train Better Responses
AI Chatbot
Your Website
Your Documents
Your Text
Your FAQ
Fine Tuning
Basically when the AI chatbot acts in a certain way you don't want it to act, you then write a sentence correcting the behavior and add it into the base system prompt.
Knowledge Base & Data Storage
This explains the technical side of how the knowledge base is created when you upload information to a chatbot.
AI Conversation A
The User's Message
User A
The AI's Response
AI Conversation B
The User's Message
User B
The AI's Response
AI Conversation C
The User's Message
User C
The AI's Response


AI Chatbot Training Better Responses Explained
How to optimize the way the AI chatbot acts/ responds in relation to certain questions from users.
Custom Knowledge Base
All the information the AI bot is trained on. It will only respond based off of the information you give it.
The Info it uses to generate responses
New Session w/ AI Chatbot
The User's Message
User starts new AI chat
The User's message is sent to the AI chatbot
AI Chatbot
Users
How the AI Chatbot Acts
Base System Prompt
This is how you fine-tune your chatbot to engage with users in certain ways, ask specific questions and respond with answers based on conditional logic.
The AI Chatbot will generate a response based on the context of the user's question + the information in the custom knowledge base WHILE making sure to obey all of the "rules" in the Base System Prompt.
The AI's response is sent to the user in the chat window.
The AI's Response
Fine Tuning
Basically when the AI chatbot acts in a certain way you don't want it to act, you then write a sentence correcting the behavior and add it into the base system prompt.


Custom Knowledge Base for AI Chatbot Explained
Your Client's Information for Their AI Chatbot
Your Text
Your Documents
Your Website
Your FAQ
Stammer.ai Knowledge Base & Data Storage
Create Text Embeddings
Parse All Information uploaded so it's able to be uploaded and retrieved from a database.
Vector Database
Add all of the text embeddings into a database that can be searched using natural language.
NLP Context Matching
The system will use the text embeddings of the user's message and search the Vector Database for related data.
Create Text Embeddings for the Users Message
The User's Message
User C
The AI's Response
Prompt Sent to OpenAI
The Contextual Search Matches
+
Top User Q/A
+
Base System Prompt