Department for
Science, Innovation
& Technology

Government
Digital Service

AI Insights

Generative AI

Introduction

Generative AI is a subset of AI capable of generating text, images, video or other forms of output by using probabilistic models trained across one or more domains. These models learn from large amounts of specially-curated training data to discern and replicate complex patterns and structures. The output generated by these models mimics the characteristics learned from that training data, enabling many novel applications – ranging from personalised content generation, to advanced analysis and evaluation, and aiding creative processes.

How generative AI works

The general functioning of generative AI models can be understood by looking at the example of how large language models (LLMs) process data to produce output. The same high-level processes are followed for speech, audio, music, video and other outputs: the data and media change, but the pattern matching, recognition and probabilistic output generation remain largely the same.

An LLM begins by first tokenising the input context of words, effectively extracting sub-words (tokens) from the text making them more easily processed by the system. Each token is then converted into an embedding. Computer systems can only process numbers. These embeddings are the numbers that generative AI uses to process language. More specifically, they are vectors of numbers which capture associated semantic information. The embeddings are subsequently associated with a positional encoding, so that the original order of words is maintained for the LLM.

The combined embeddings are then fed through a series of transformers, comprising attention mechanisms and feed-forward neural network operations that allow the model to generate context-specific output.

While this process enables LLMs to produce responses to requests, these models do not understand the content or meaning of the words beyond how an input set of numbers may most likely translate into an output set of numbers, and subsequently into streams of words. This means that there are limitations on what these models can reasonably do. You must consider that these are probabilistic models, and, despite producing human-like output, they are not at all sentient.

Core concepts

Some concepts used in designing and building generative AI solutions are:

- prompts: the primary input provided to an LLM. In the simplest case, this is the query created by a user. In production systems, a prompt will usually have additional parts, such as system prompts, safeguards, context, history, and context-specific data

- prompt engineering: this describes the process of adjusting LLM queries to improve quality and accuracy.
- user prompt: this is the underlying user query. User prompts are generally written in everyday natural language
- system prompts: higher-level instructions that help direct an LLM to respond in a specific way. They can be used to instruct the model on how to generate responses to user prompts, provide feedback or handle certain types of content
- inference: the process of making predictions, or generating outputs, from a machine learning or other AI model
- retrieval augmented generation (RAG): techniques that allow for the inclusion of selected data in a prompt. This is especially useful, and is used to make that data available to the LLM to consider when responding to a user query
- vector databases: these are databases that specialise in storage and retrieval of data in numerical formats called embeddings. These help in quickly finding documents similar in content by comparing these numerical representations
- grounding: the process of linking the representations learned by the AI models to real-world entities or concepts provided. This allows the model to incorporate data beyond its training data to formulate responses
- open-source models: these are publicly accessible models. Their source code, architecture and parameters are available for examination and modification by the broader community. Sites such as HuggingFace are directories containing open-source models and data sets
- closed-source models: these are proprietary and the inner workings and details of these models are kept confidential and are not shared publicly

Model deployment

Generative AI can be deployed in different ways. Each deployment pattern offers different benefits and presents a different set of security challenges, affecting the level of risk that must be managed. This section describes the ways in which you're likely to use and encounter generative AI, including:

- public generative AI applications and web endpoints
- integrated generative AI applications
- local development
- cloud solutions

Public generative AI applications and web endpoints

Vendors often make their generative AI systems freely available for evaluation, under licence, by presenting a user interface where queries can be entered and responses viewed. Usually the context and session history are maintained so a user may ask follow-up questions, and see previous interactions over time.

Vendors may also allow registered users to access similar services over an application programming interface (API). This allows the user to call the LLM from within their own software applications, typically by connecting to a well-known URL and providing an API key.

There are a few things you'll need to consider before signing up to generative AI services for work purposes:

- you must make sure you're acting in line with the policies of your organisation
- while the use of these services may be free of charge, you should be aware that any information provided to these services may be used by the provider
- free-tier services often have restrictions that prevent their use beyond evaluation purposes
- data may be stored, even temporarily (such as log data), outside of the UK jurisdiction
- the use of the output might be governed by a set of conditions set by the vendor
- generative AI services and applications are vulnerable to bias in their training data, as for any model. You're responsible for checking and controlling their output
- they may also produce unusual results, and this should be expected as normal and protected against. Any generative AI system should be profiled to evaluate the response behaviour and ensure it's in line with expectations

Integrated generative AI applications

Generative AI is becoming increasingly integrated into mainstream products. Integrated generative AI tools provide straightforward user interfaces in products that people are already familiar with. They can be a very simple way to bring the benefits of generative AI into an organisation, because these tools allow people to use language-based prompts to ask questions about their organisation's data, or for specific support on a task.

Examples of integrated generative AI tools include but are not limited to:

- Adobe Photoshop Generative Fill tool: this helps with image editing by adding or removing components
- AWS ChatOps Chatbot: an AI assistant that can help manage an Amazon Web Services (AWS) cloud environment
- Microsoft 365 Copilot: an AI assistant that can support the use of Microsoft products
- Google Duet AI: an AI assistant that can support the use of Google products, including writing code

You must understand the data processing and maintenance of these services before using them. For example, some products that provide services such as abuse monitoring may still retain information for processing by the vendors. If data sovereignty is a concern, you must also clarify the data processing profile with the vendor. This includes geolocation, data ownership and usage.

LLMs that are integrated into existing enterprise licences may have widespread access to the organisation's data. Before enabling a service, you must understand what data is visible to integrated AI services, and how that data is consumed, processed, and potentially transmitted or communicated externally.

Updates to these systems tend only to increase their harvesting capability and complexity. You should not assume that only local data is accessible: any API, endpoint, website or database available to the user may be reachable. Any data the user has access to may therefore be accessed by generative AI processes run by those users. The same concern applies to server-installed processes running under service accounts.

Local development

Many generative AI development scenarios begin with prototyping or minimum viable product assessments on a local or near-local setup. This means directly on a development workstation, or coding against a generative AI instance running on the local area network.

Usually these local instances are not the fully featured versions that are deployed locally or remotely in production. Smaller models, such as 7- or 8-billion parameter instances, may be used for pipeline and parameter tuning and rapid turnaround. These instances are subject to the same vulnerabilities as their larger counterparts. Equally, it's not sufficient to develop on smaller models, and then deploy on larger ones, without executing not only the full test suite against them, but also tests specific to the capabilities of the larger model. Profiling and attack-surface evaluation testing are also mandatory prior to any model change. This means that the test suite should be a function of the size of the model and the deployment target.

Cloud solutions

Cloud services provide similar functionality to public and paid-for APIs, often with a familiar web UI. In addition to compliance with the Government Cloud First policy, cloud services allow increased control over your data.

When establishing your generative AI cloud service, make sure the development environment complies with your organisation's data security policies, as well as with legal and governmental guidelines and the government Service Standard.

You should map a formal process to decide the appropriate cloud vendor, tooling and model based on the use case that you're addressing. If your organisation and/or use case requires all data to remain within the UK, you might need to factor in additional time to apply for access to resources within the UK, as these may be subject to additional regulation by some providers. Technical account managers and solution architects supporting your enterprise account will be able to help with this step.

Tools and technology

Using the right tools and technology for the task is integral to successfully developing and deploying generative AI. Your current IT infrastructure, level of expertise, risk appetite and the specific use cases you propose will all inform the correct selections.

You'll need to consider a number of technology choices when building your generative AI solutions, including the most appropriate IT infrastructure.

Infrastructure

You should select a suitable infrastructure environment. Microsoft, Google or AWS may be appropriate depending on your current IT infrastructure, or existing partnerships and expertise in your teams.

Alternatively, a specific foundation model may be more appropriate for your particular use case, leading to a particular set of infrastructure requirements.

You should consider that, as models change and improve, the most appropriate model for your use case may also change. An extensible and modular architecture, which allows for swapping pipeline components including the target model, is highly recommended.

You should also consider:

- cloud services versus local development: you should be aware of the UK government's Cloud First policy, but understand that local development may be feasible for experimentation
- container technology: using this from the start can help you to move your solution between platforms with minimal overhead
- data storage: vector stores are often a vital component in RAG systems, and you should evaluate these for suitability
- access logging, prompt auditing and protective monitoring

Frameworks

Generative AI frameworks provide tools, APIs and pre-built models to develop, train and deploy generative AI models.

These frameworks have their own uses, strengths and unique features. They may also contribute to vendor lock-in – so again, a modular architecture is recommended.

The choice of a generative AI framework depends on:

- your specific project requirements
- the familiarity of your team with the framework and programming language
- the size and level of engagement of the community support around it

Model selection

There is an ever-increasing number of models available. Selecting the correct one is extremely important. Choosing a model is arguably more challenging now, when there are many competing, overlapping alternatives, than before when there were few realistic alternatives to select from.

Here are some of the things you should consider:

- data security: this is the top priority of any public sector data processing system, whether behind the firewall or not, and regardless of authorisation or privilege levels. You must ensure that systems are secure, and that data sandboxing, anonymisation, and safeguarding form part of any delivery
- data manipulation: foundation models provide an access route to data and behaviour that was previously locked down behind very specific and limited APIs. The ability to generally query models through text, images, video and other forms of output may also expose attack vectors that organisations are not familiar with. ASCII Art attacks, image-embedding attacks, many-shot or zero-shot jailbreaking, among many others, form novel, significant

vulnerabilities. The cost of training many of these models, in time and monetary terms, means that traditional patching processes are not viable. Constant vigilance is required

- data selection: selecting data for training models, or to augment model behaviour, is vital. An old adage in computer science is garbage in garbage out (GIGO). If you train your model on mismatched data, or make poor data selections available to a model, you can expect a poorer quality in the responses that are output
- model selection: selecting an appropriate model is also very important. The best quality data will suffer if pushed through a model which cannot correctly ingest and extract based on that data. Models have individual strengths and weaknesses – it's important that you know these, and evaluate and compare them to system objectives
- cost: most access to LLMs is charged by the number of tokens (effectively sub-words, as described in the How generative AI works section). This runtime cost is often overlooked during a costing and budgeting phase. Typically, there are also 2 associated charges. The first is for inbound tokens in the request, which have to be parsed and processed. The second is for outbound tokens in the response. Sending free text to the LLM can lead to uncontrolled cost. Controlling the query and caching results can help to manage and monitor the overall cost at runtime
- open-source versus closed-source models (training data, code and weights): if openness is important to your solution, you should consider whether all aspects of the model – including training data, neural network coding and model weights – should be available as open source

Getting reliable results

There are several things you can do to help deliver high quality and reliable performance in generative AI systems:

- protect your data and systems from deliberate or inadvertent use or attack
- select your model carefully: to achieve a reliable, consistent and cost-effective implementation, choose the most appropriate model for a particular use case
- organisation data**:** enabling the model to access and use organisation data can provide insights and knowledge to users that is specific to their subject domain. There are different ways to hook a generative AI model into these data sources:

    - model training: models may be trained from scratch on organisation data. While this requires a higher level of expertise, it can produce more specific model behaviour that brings performance and security benefits
    - RAG: Implementing retrieval augmented generation involves matching user queries with relevant organisational data and providing that information to the LLM to generate informed responses
    - fine-tuning: taking a pretrained model and refining the model weights by training it again on specific subsets of organisational data
- evaluate input prompts: you can evaluate user inputs to the generative AI tool through safeguards to detect and filter inappropriate inputs. Example checks include:
    - identifying whether the prompt is abusive or malicious
    - confirming the prompt is not attempting to jailbreak the LLM, for example by asking the LLM to ignore its safety instructions
    - confirming no unnecessary personally identifiable information (PII) has been entered
- use prompt engineering techniques to tune your prompts. See our Prompt Engineering Insight
- evaluate outputs: once a model returns an output, it's important to evaluate the response. Depending on the use case, a human might be required to check the output some or all of the time. You must select the correct metrics in order to quantify responses and evaluate the model behaviour, as this is also vital for automated testing and comparison
- include humans: the 2 main mechanisms for human involvement and oversight are human-in-the-loop (HITL), and reinforcement learning from human feedback (RLHF):
    - HITL systems involve people to review, correct and approve system output. They're a vital aspect of delivering critical services, especially where the cost of failure may be high
    - RLHF systems are reward-based, and are used to align a system with preferences and values monitored by a human operator

Testing generative AI solutions

Generative AI systems are fallible and the correctness of their responses is not guaranteed. These systems are probabilistic models which predict the likeliest

outputs for given inputs. They generate responses that have a high measure of plausibility based on the data that they have processed. This means that they can, and do, make errors. In addition to employing techniques to get reliable results, you should have a process in place to test and measure them:

- conduct thorough testing to assess the functionality and effectiveness of the system
- record the input prompts and the returned responses, and collect and analyse metrics across all aspects of performance – including hallucinations, toxicity, fairness, robustness, and higher-level business key performance indicators
- evaluate the collected metrics and validate the model's outputs against ground truth or expert judgement, obtaining user feedback if possible
- closely review the outcomes of the technical decisions made, the infrastructure and running costs and environmental impact. Use this information to continually iterate your solution

There are many evaluation metrics for assessing performance in generative AI. There are also tools you can use, for example to support AI fairness testing, including:

- IBM fairness 360
- Microsoft FairLearn
- Google What-If-Tool
- University of Chicago Aequitas tool
- PyMetrics audit-ai

Please take a look at our LLM Testing Insight.

Generative AI data management

Good data management is crucial to supporting the successful implementation of generative AI solutions. The types of data you'll need to manage include:

- testing and operational data: all model inputs and outputs should be logged during development and testing. This information will be used to improve the performance of the system. It should be possible to audit this information in production, monitor and maintain performance and investigate anomalies
- financial data: the cost of running your generative AI solutions should be well known, and monitored closely. This will ensure you continue to operate cost-effectively for the given model and prompts. This relates to pure input

and output token charges, and also to the hosting environment itself, hosting servers, data stores, vector databases.

Synthetic data

Data that is scarce or hard to obtain may be supported through the synthetic generation of data. Synthetic data is created to look like and mimic the statistical distribution of real data. For instance, if you were trying to build a fraud detection system but did not have many examples of real fraudulent activities, you could use synthetic data. This is also a technique widely used in medical data, where generative methods are used to standardise image data from multiple incompatible sources.

As always, there are caveats. Synthetic data will have well-known distributions built into it, and no more. If the original distributions are incorrect, or incomplete, then the model will inherit those deficiencies. Synthetic data is intended to extend your capacity and capabilities. You should pay the same care and attention to synthetic data generation as to the data pipeline and model selection.

We have more detail on this subject in our Synthetic Data Spotlight.

Final Thoughts

This document serves as an introductory overview or refresher to the fundamental concepts and considerations associated with generative AI. It is important to recognise that this field is vast and rapidly evolving. Staying informed about the latest developments is crucial, as new capabilities and challenges emerge regularly. Engaging with experts, participating in workshops, and reviewing additional resources can further enhance your knowledge.