A PRACTICAL GUIDE TO BUILDING GENERATIVE AI APPLICATIONS

Contents

Introduction to Generative AI

Overview

Generative AI refers to a subset of artificial intelligence that focuses on creating new content and data rather than analysing or acting upon existing data. Unlike traditional AI, which primarily processes and interprets information, generative AI can produce novel text, images, music, and even entire simulations based on patterns and inputs it has learned during its training. The core technologies enabling generative AI include advanced machine learning models, particularly neural networks such as Generative Adversarial Networks (GANs) and Transformer-based models like GPT (Generative Pre-trained Transformer). (1)

Generative AI models are trained on vast datasets and learn to understand the underlying structures and patterns within the data. Once trained, these models can generate outputs that mimic the style, content, and context of the training data, often with exceptional coherence and creativity.

Importance and Benefits for Business

Generative AI holds transformative potential for businesses by providing them with advanced tools and capabilities. Here are some ways in which generative AI can benefit businesses:

- **Enhanced customer engagement**: Generative AI can power intelligent chatbots and virtual assistants that provide personalized, instant customer service and support. These AI systems can handle a wide range of customer inquiries, offer product recommendations, and resolve issues efficiently, leading to improved customer satisfaction and loyalty. (2)
- **Cost efficiency and scalability**: By automating repetitive and time-consuming tasks such as content creation, data entry, and report generation, generative AI allows businesses to reduce operational costs and reallocate human resources to more strategic activities. This not only increases efficiency but also enables businesses to scale their operations without a proportional increase in costs. (3)
- **Creative content generation**: Generative AI can assist in creating high-quality marketing materials, including text, images, and videos. AI-driven content generation tools can produce engaging social media posts, articles, and promotional materials. (2)
- **Data-driven decision making**: Generative AI can analyse large datasets to generate insights and predictive analytics, providing businesses with valuable information for decision-making. By understanding customer behaviour, market trends, and operational performance, businesses can make informed decisions that drive growth and competitiveness. (4)
- **Innovation and competitive advantage**: Generative AI can assist in the ideation and design of new products by generating innovative concepts and prototypes. This accelerates the development process and helps bring new products to market faster, maintaining a competitive edge. (2)

Use Cases of Generative AI

Generative AI has the potential to transform various aspects of business by automating processes, enhancing creativity, and improving customer engagement. Here are some of the use cases of generative AI (Source: (5)):

- Marketing content assistant
- Code assistant for developers
- Customer support on demand
- Product design assistant
- Research-based report generation
- Synthetic data generation
- Enterprise-wide data search and access
- Game content development
- Language translation
- Simulation of urban planning scenarios
- Hyper-personalised education
- Onboarding assistant.

Components of a Generative AI Application

Before diving into specific tools and frameworks, it's important to understand the different components that make up a generative AI application. These components work together to create sophisticated AI systems capable of generating content, answering questions, and more.

1. **Large Language Models (LLMs):** LLMs are deep learning models trained on vast amounts of text data. They can generate coherent and contextually relevant text based on given prompts. Examples: GPT-4, Gemini, Llama 3, Claude 3, Poro, Gemma.
2. **Knowledge Bases:** A knowledge base is a centralized repository that stores information, enabling easy access and retrieval for various applications. It supports the integration and retrieval of data necessary for generative AI systems to function effectively.
    a. Vector databases (Pinecone, Weaviate): Store data as high-dimensional vectors, allowing efficient similarity searches and semantic queries.
    b. Relational databases (MySQL, PostgreSQL): Store structured data in tables with defined relationships, ideal for complex queries and transactions.
    c. Document stores (MongoDB): Manage unstructured data in document formats (e.g., JSON), offering flexible schema and easy scalability.
    d. Graph Databases (Neo4j): Represent data as nodes and edges, suitable for exploring relationships and connected data.
    e. In-Memory Databases (Redis): Store data in memory for rapid access and real-time processing, beneficial for high-speed transactions.
3. **Embedding Models:** These models convert data into numerical vectors that capture semantic meaning, facilitating tasks like similarity search and clustering. Examples:

OpenAI embeddings (text-embedding-3-large), Cohere embeddings (embed-multilingual-v3.0), Google embeddings (text-multilinguel-embedding-preview-0409).

4. **Backend Logic:** The backend logic integrates various components and handles the business logic, API calls, data processing, and more. It ensures that the application functions smoothly and efficiently. Examples: Flask, Node.js.
5. **Orchestration and Workflow Tools:** These tools help in chaining together different AI models and services to create end-to-end applications. They manage the flow of data and tasks between components. Examples: Langchain, LlamaIndex.
6. **User Interface:** The front-end component that interacts with users, providing an accessible and user-friendly way to interact with the AI application. Examples: React, HTML/CSS, Javascript.

The diagram illustrates the flow and components of a generative AI application, detailing how different elements interact to process user inputs and generate outputs (Figure 1).

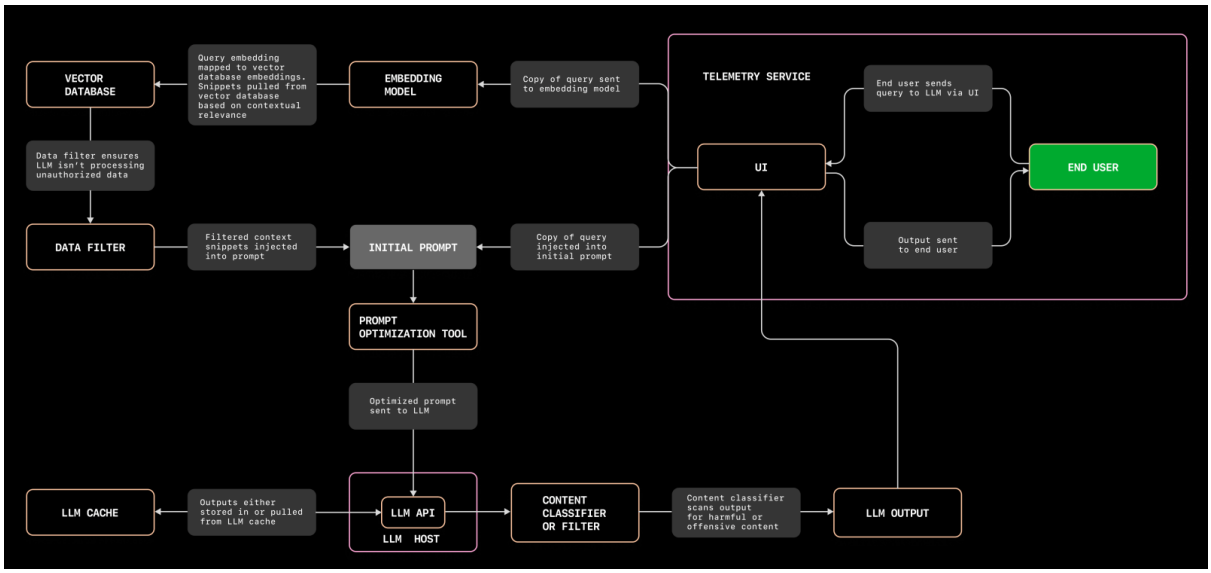

Figure 1. The architecture of an LLM application showing the major components. (Source: (6))

Choosing the Right Generative AI Tools and Frameworks

Generative AI applications rely on various tools and frameworks to function effectively. These tools manage different aspects of the application, from data processing, orchestration, and response generation. Selecting the right tools for your generative AI applications involves understanding the specific needs of your project.

Large Language Models

When choosing a generative AI model for your business needs, it is essential to consider the strengths, weaknesses, and specific capabilities of each model. Here are some of considerations (source: (7)):

1. Use case specificity: Determine the primary task, such as text generation, coding assistance, or comprehension. For instance, GPT-4 is highly versatile, suitable for a wide range of applications, while specialized models like Code LLaMa are optimized for programming tasks.
2. Interfacing methods:
    a. Playground interfaces are simple to use but limited in customization (e.g. ChatGPT, Google's Gemini)
    b. API access allows for integration into applications without infrastructure management but incurs costs (e.g. OpenAI's GPT-4 API is easy to integrate)
    c. Hosting locally offers full control and customization but requires significant technical effort and resources (e.g running LLaMa models using Ollama).
3. Performance and cost:
    a. Compare performance against cost, especially for extensive usage. GPT-3.5 models are cheaper than GPT-4, making them a viable option for less demanding tasks.
    b. Higher parameter models like GPT-4 require more computational power, impacting costs and deployment strategies.
4. Customization and control:
    a. If specific tuning is required, models like OpenAI's fine-tuned GPT models or Meta's LLaMa with instruction tuning can be beneficial.
    b. Hosting models like LLaMa locally provides customization but demands substantial infrastructure management.

Hugging Face is a leading platform in the AI community, providing tools and resources for developing, deploying, and sharing machine learning models. It is particularly known for its extensive repository of pre-trained models for natural language processing (NLP) and other AI tasks. Hugging Face facilitates AI development and deployment, making advanced models more accessible.

Spotlight on Poro Model

Poro 34B is an open-source, multilingual large language model developed through a collaboration between Silo AI, the TurkuNLP group at the University of Turku, and the High Performance Language Technologies (HPLT) project. Trained on the LUMI supercomputer, Poro 34B excels in English, Finnish, and code. (8)

Vector Databases

A vector database is a specialized database optimized for storing and querying high-dimensional vector data, commonly used in machine learning and AI applications. Vectors represent data in a numerical format that captures semantic meaning, making them ideal for tasks like similarity search, recommendation systems, and natural language processing.

Vector databases operate by transforming data into vector embeddings, which are high-dimensional representations of the data. These embeddings are stored in the database and can

be quickly queried using various algorithms to find the nearest neighbours or most similar items. The process typically involves:

1. Raw data is converted into vector embeddings using machine learning models.
2. The embeddings are stored in the database, often with additional metadata for filtering and enhanced search capabilities.
3. When a query is made, the database uses Approximate Nearest Neighbour (ANN) algorithms to find and return the most relevant vectors quickly. (9)

Selecting a vector database involves evaluating several key factors to ensure it meets your specific use case requirements (Source: (9)):

- Performance:
    - Data freshness: Ensure the database can quickly update and query new data to maintain relevance, crucial for real-time applications.
    - Query latency: Look for low query response times to support interactive applications like chatbots and search engines.
    - Query per second (QPS): High QPS capabilities are essential for handling large volumes of queries efficiently.
- Relevance:
    - Accuracy: Balance speed and accuracy by selecting a database with robust algorithms for approximate nearest neighbour searches.
    - Hybrid search: Consider databases that support both semantic and keyword search for more versatile querying.
    - Metadata filtering: Ability to filter queries based on metadata enhances search relevance and specificity.
- Scalability:
    - Scaling capabilities: Assess whether the database can scale vertically (adding more resources to existing servers) and horizontally (adding more servers) to handle growing data volumes.
    - Automatic scaling: Prefer databases that support automatic scaling to simplify management.
- Cost-efficiency:
    - Cost performance: Evaluate the cost against the performance benefits, especially as you scale up the number of embeddings.
    - Predictable pricing: Look for transparent pricing models to avoid unexpected costs.
- Developer experience:
    - Ease of use: Check for user-friendly APIs and SDKs to facilitate quick integration and development.
    - Documentation: Comprehensive and clear documentation helps developers get started and troubleshoot issues effectively.
    - Integrations: Ensure the database integrates well with your existing tech stack and preferred cloud providers.
- Enterprise readiness:

    - Security and compliance: Verify that the database meets necessary security standards and compliance requirements (e.g., SOC 2, GDPR).
    - Support and expertise: Consider the availability of technical support and expertise from the provider to assist with deployment and scaling.
    - Monitoring and availability: Robust monitoring tools and high availability are critical for maintaining uptime and performance.
- Database management and hosting:
    - Managed services: Some vector databases are fully managed by the provider, hosted either on their platform or the client's cloud, offering ease of use and reducing the need for in-house management.
    - Self-managed options: Others can be managed by the user, providing flexibility to host on cloud or local infrastructure, which may require more technical expertise and resources.

Examples of vector databases (Source (10)):

- Vector libraries (e.g., FAISS, HNSWLib, ANNOY): Suitable for static data, lacking CRUD (Create, Read, Update, Delete) support.
- SQL databases with vector support (e.g., pgvector, Supabase): Convenient for integrating vector data but less efficient for large-scale vector similarity searches.
- Full-text search databases (e.g., ElasticSearch, OpenSearch): Excellent for text search but underperform in vector similarity searches.
- NoSQL databases with vector support (e.g., Redis, MongoDB): Basic vector capabilities, not optimized for vector similarity search.
- Dedicated vector databases (e.g., Pinecone, Milvus, Weaviate, Qdrant): Designed for vector operations, supporting high query workloads and advanced indexing algorithms.

For a continuously updated comparison of vector databases, refer to the comprehensive list provided by Superlinked, available at https://www.superlinked.com/vector-db-comparison.

Embeddings

Embeddings are numerical representations (vectors) of data such as text, images, or audio. These vectors capture the semantic meaning of the data, enabling efficient retrieval and comparison. In natural language processing (NLP), embedding models transform text into vectors, allowing similar texts to be placed close together in vector space.

Retrieval-Augmented Generation (RAG) relies heavily on embeddings to retrieve relevant information from knowledge bases, enhancing the context and accuracy of generated outputs.

Evaluate the model using benchmarks like the MTEB Leaderboard on Hugging Face, which provides performance statistics across various tasks.

- Focus on metrics like Retrieval Average (or Normalized Discounted Cumulative Gain (NDCG)), which measures retrieval accuracy.
- Larger models typically offer better performance but come with increased computational requirements and latency. Balance the need for performance with available resources.

- Check the maximum number of tokens the model can process in a single embedding. Generally, a limit of 512 tokens is sufficient for most applications.
- The length of the embedding vector affects both performance and storage. Shorter vectors are faster and more storage-efficient, while longer vectors capture more data nuances. Aim for a good trade-off based on your application's needs. (11)

LLM Orchestration

LLM Orchestration involves managing and optimizing large language models (LLMs) to enhance their performance and effectiveness. It includes tasks such as generating effective prompts, combining outputs from multiple LLMs, efficiently managing resources, and monitoring performance.

LLM orchestration increases developer productivity by automating complex tasks, improves application performance through optimized resource usage, reduces development costs, and enhances scalability and reliability. This structured approach allows to focus on core functionalities, making applications more robust and efficient. (12)

LangChain and LlamaIndex are two prominent LLM orchestration frameworks. Each offers unique features and benefits suited to different use cases (Source: (13)):

- LangChain is designed to simplify the creation and management of complex workflows involving multiple AI models and services. It focuses on integrating LLMs with various data sources and orchestrating their interactions.
    - Key features:
        - LangChain provides a high-level API that allows to define intricate workflows and chains of AI models and services.
        - It integrates well with existing tools and services, making it adaptable to a wide range of applications.
        - The framework excels at managing prompts, ensuring that they are effectively used to elicit the desired responses from LLMs.
    - LangChain is ideal for building sophisticated applications such as advanced chatbots, virtual assistants, and AI-driven content creation tools that require integrating multiple AI models and external data sources.
- LlamaIndex is focused on bridging the gap between powerful LLMs and private or domain-specific data. It provides a structured way to ingest, organize, and query various data sources using LLMs.
    - Key features:
        - LlamaIndex allows for seamless integration with diverse data sources such as APIs, databases, and document files.
        - Facilitates natural language interactions with private data without the need to retrain models.
        - Offers both high-level and low-level APIs to cater to different expertise levels among developers.
    - LlamaIndex is perfect for applications that require integrating LLMs with specific datasets, such as internal knowledge bases, proprietary documents, and

specialized databases. It excels in scenarios where natural language querying of private data is essential.

When deciding between LangChain and LlamaIndex, the specific needs of the project need to be considered:

- If the application requires sophisticated orchestration involving multiple AI models and services, LangChain's comprehensive tools and flexibility make it the better choice.
- For projects focused on integrating LLMs with private or domain-specific data sources, LlamaIndex offers a more straightforward and efficient solution.

Both frameworks offer powerful capabilities, but their suitability depends on the complexity and specific requirements of the generative AI application.

Risks and Challenges of Generative AI and Their Mitigation Strategies

While generative AI offers numerous benefits, it also comes with its own set of risks and challenges that businesses need to consider. Understanding these challenges is crucial for effectively implementing and leveraging generative AI technologies.

Unpredictable Output

Unpredictable output refers to the inherent variability in the quality and appropriateness of content generated by generative AI models. These models can produce a wide range of outputs, some of which may not align with brand guidelines, cultural norms, or the intended use case. (14)

Mitigation Strategies

Human-in-the-Loop Review

Implementing a robust human review process is essential to assess the quality and appropriateness of AI-generated content. Human reviewers can provide the necessary context and judgment that AI models lack.

Guideline Enforcement

Establishing and enforcing strict brand and content guidelines can help mitigate risks. Clear guidelines on tone, style, and cultural sensitivity can aid human reviewers and AI systems in producing consistent and appropriate content.

Feedback Loops

Implementing feedback mechanisms where users and stakeholders can report inappropriate or low-quality outputs allows for continuous improvement of the AI models. This feedback can be used to improve future outputs.

Hallucinations

AI hallucinations refer to instances where generative AI models produce information that appears plausible but is incorrect or nonsensical. These models generate content based on patterns in the data they were trained on, but when they encounter gaps in their knowledge or ambiguous prompts, they may fabricate information to fill in those gaps. (15)

Hallucinations can lead to the dissemination of false or misleading information, which can significantly harm a company's credibility and decision-making processes. Hallucinations pose a particular risk in scenarios where accuracy and reliability are critical. They can undermine the perceived reliability of AI systems, necessitating rigorous human oversight to verify the correctness of AI-generated outputs. Without proper mitigation, the risk of hallucinations can outweigh the benefits of using generative AI. (16)

Mitigation Strategies

Retrieval-Augmented Generation (RAG)

Integrating information retrieval with text generation can help reduce hallucinations. By retrieving relevant documents or data from a verified corpus before generating responses, RAG ensures that AI outputs are based on accurate and relevant information, enhancing their reliability. (15)

Human-in-the-Loop Review

Implementing a system where human reviewers verify AI-generated content can significantly reduce the risk of hallucinations. Humans can cross-check the information provided by the AI against reliable sources, ensuring accuracy before the content is published or used. (16)

Training on High-Quality Data

Ensuring that AI models are trained on high-quality, accurate, and diverse datasets can help reduce the frequency of hallucinations. High-quality training data enables the models to generate more reliable and contextually appropriate content. (16)

Prompt Tuning

Prompt tuning involves customizing prompts to guide large language models towards producing desired outputs. This approach helps avoid extensive retraining by providing carefully designed prompts that lead to more accurate and contextually appropriate responses. (15)

Here are several prompting tips:

- Ask the LLM to explain how it arrived to the conclusions or to provide evidence to support its claims;
- Provide the LLM with output examples or a response structure;
- Break down a complex task to smaller steps;

- Be more specific, do not leave room for ambiguity. (17)

Data Privacy and Security

Generative AI systems often require large amounts of data to function effectively. Handling and processing this data raises significant privacy and security concerns, particularly when dealing with sensitive or personal information. Ensuring that data is protected from unauthorized access and breaches is vital. (18)

Inadequate data protection measures can lead to severe consequences such as data breaches, legal repercussions, and loss of customer trust. For businesses, a data breach can be particularly damaging, leading to financial losses, regulatory fines, and a tarnished reputation.

Mitigation Strategies

Compliance with Regulations

Complying with data protection regulations such as the General Data Protection Regulation (GDPR) is essential. These regulations provide stringent requirements and guidelines for maintaining data privacy. (19)

The new EU AI Act categorizes AI applications by risk levels, imposing strict requirements on high-risk AI systems to ensure transparency, data privacy, and security. Compliance with this Act will be crucial for businesses operating within or engaging with the EU market. *(20)*

Access Controls

Implementing robust access control mechanisms limits access to generative AI models and the data they generate to only authorised personnel. (19)

Data Anonymization

Removing personally identifiable information before feeding it to generative models protects individual privacy. (19)

Secure Data Storage

Utilizing secure cloud storage solutions with robust security measures such as encryption, regular security updates, and secure access protocols safeguards data. (19)

User Consent

Informing users of the intended use of their data and requesting their consent, this way ensuring a transparent system and an informed decision-making process by the users. (19)

Data Governance Policies

Implementing a comprehensive data governance framework establishes clear policies and procedures for data management, including data handling and protection standards. (19)

Bias and Fairness

Bias in generative AI occurs when models produce outputs that reflect prejudiced or unfair perspectives, often derived from the biased data on which they were trained. Fairness refers to the need to ensure that AI systems treat all users and scenarios equitably, without discrimination or favouritism. (21)

Bias in AI can lead to unfair treatment of individuals or groups, perpetuating inequality and potentially causing harm. For instance, biased AI outputs can affect hiring decisions, customer service interactions, and product recommendations, leading to legal and reputational risks for businesses. Ensuring fairness is critical to maintaining trust and promoting ethical AI use. (22)

Table 1. Bias in AI systems. (Source: (21))

Type of bias	Description	Examples
Sampling	Occurs when the training data is not representative of the population it serves, leading to poor performance and biased predictions for certain groups.	A facial recognition algorithm trained mostly on white individuals that performs poorly on people of other races.
Algorithmic	Results from the design and implementation of the algorithm, which may prioritize certain attributes and lead to unfair outcomes.	An algorithm that prioritizes age or gender, leading to unfair outcomes in hiring decisions.
Representation	Happens when a dataset does not accurately represent the population it is meant to model, leading to inaccurate predictions.	A medical dataset that underrepresents women, leading to less accurate diagnosis for female patients.
Confirmation	Materializes when an AI system is used to confirm pre-existing biases or beliefs held by its creators or users.	An AI system that predicts job candidates' success based on biases held by the hiring manager.
Measurement	Emerges when data collection or measurement systematically over- or underrepresents certain groups.	A survey collecting more responses from urban residents, leading to an under-representation of rural opinions.
Interaction	Occurs when an AI system interacts with humans in a biased manner, resulting in unfair treatment.	A chatbot that responds differently to men and women, resulting in biased communication.
Generative	Occurs in generative AI models, like those used for creating synthetic data, images, or text. Generative bias emerges when the model's outputs disproportionately reflect specific attributes, perspectives, or patterns present in the training data, leading to skewed or unbalanced representations in generated content.	A text generation model trained predominantly on literature from Western authors may over-represent Western cultural norms and idioms, under-representing or misrepresenting other cultures. Similarly, an image generation model trained on datasets with limited diversity in human portraits may struggle to accurately represent a broad range of ethnicities.

Mitigation Strategies

Diverse and Representative Training Data

Ensuring that training datasets are diverse and representative of all relevant user groups and scenarios. This helps the AI model learn a more balanced perspective. (22)

Human-in-the-Loop (HITL) Approaches

Establishing robust feedback loops where users can report biased or unfair outputs. This feedback can be used to continuously improve the AI model. (22)

Transparent and Explainable AI

Ensuring transparency in how AI models are trained and make decisions. Clear documentation and communication about the model's workings can help users understand and trust the AI system. Additionally, developing explainable AI models that can provide insights into how decisions are made. This helps in identifying and addressing potential biases. (21)

Technical Complexity

Technical complexity in generative AI refers to the challenges involved in implementing, maintaining, and optimizing AI systems. High technical complexity can result in increased costs, longer implementation times, and potential failures in deploying AI solutions effectively. Business may struggle with limited technical expertise and resources, hindering their ability to leverage AI fully. (14)

Mitigation Strategies

User-Friendly Tools and Platforms

Leveraging AI tools and platforms that offer user-friendly interfaces and require minimal technical expertise. These tools often provide pre-built models and easy-to-use frameworks that simplify AI implementation. Additionally, utilizing managed AI services from cloud providers that handle the underlying infrastructure and maintenance can significantly reduce technical complexities associated with AI applications.

Partnerships and Collaboration

Collaborating with AI experts, consultants, or technology partners who can provide the necessary technical support and guidance as well as engaging with academic institution to stay updated on the latest AI advancements and best practices. This can provide access to cutting-edge research and technical expertise.

Incremental Implementation

Implementing AI solutions in phases, starting with proof-of-concept projects and gradually scaling up. This allows businesses to learn and adapt to the complexities of AI without overwhelming their resources.

Training and Skill Development

Investing in training programs to upskill employees on AI technologies and best practices. Encouraging participation in AI workshops, courses, and certification programs to enhance technical knowledge and competencies. This helps build in-house expertise and reduces reliance on external resources.

Scalable Infrastructure

Leveraging cloud computing resources to scale AI infrastructure as needed. Cloud platforms offer scalable and flexible environments that can handle the computational demands of AI.

Getting Started with Generative AI in Business

Implementing generative AI involves several steps to ensure success and value realization (Source: (23)):

1. Education and motivation: A thorough understanding of generative AI is essential before implementation. This involves reading books, articles, and taking courses to stay updated with the rapidly evolving technology.
2. Focusing on real problems: Generative AI should be applied to solve specific, minor problems initially. Starting small allows for internal testing and skill development before scaling to more significant, customer-facing projects.
3. Selecting tools and partners: Choosing the right AI tools and platforms is critical. Initially, partnering with experts can help in building a realistic AI roadmap and navigating through various available tools.
4. Creating an AI-ready culture: An AI-ready culture is characterized by employees viewing AI as an opportunity, willingness to integrate AI across functions, and focusing on augmenting human abilities rather than replacing them. Transparency, training, and incentivizing AI use cases are vital for this culture.
5. Learning prompt engineering: Prompt engineering is crucial for utilizing generative AI tools effectively. It involves crafting precise prompts to guide AI models in generating the desired outputs. Understanding and improving prompts can significantly enhance the results.

Additional Resources

- The economic potential of generative AI: The next productivity frontier *https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#introduction*
- A continuously updated list of Large Language Models that are licensed for commercial use: https://github.com/eugeneyan/open-llms
- A continuously updated detailed list of LLMs with information about parameters, training datasets and other notes: https://lifearchitect.ai/models-table/

- A continuously updated comparison of vector databases: https://www.superlinked.com/vector-db-comparison
- What is Trustworthy AI? https://blogs.nvidia.com/blog/what-is-trustworthy-ai/#:~:text=Trustworthy%20AI%20is%20an%20approach,people%20who%20interact%20with%20it.
- Why businesses need explainable AI—and how to deliver it: https://mckinsey.com/capabilities/quantumblack/our-insights/why-businesses-need-explainable-ai-and-how-to-deliver-it

References

1. What is generative AI? *IMB*. [Online] https://research.ibm.com/blog/what-is-generative-AI.

2. Ihnatchyck, Yana. Top 4 generative AI benefits for business. *Data Science Central*. [Online] https://www.datasciencecentral.com/top-4-generative-ai-benefits-for-business/.

3. Marr, Bernard. Boost Your Productivity with Generative AI. *Harward Business Review*. [Online] https://hbr.org/2023/06/boost-your-productivity-with-generative-ai.

4. Six Generative AI Benefits for Small Businesses. *Microsoft*. [Online] https://www.microsoft.com/en-us/microsoft-365/business-insights-ideas/resources/generative-ai-benefits-for-small-businesses.

5. The Generative AI Dossier: A selection of high-impact use cases across six major industries. *Deloitte*. [Online] https://www2.deloitte.com/us/en/pages/consulting/articles/gen-ai-use-cases.html#.

6. Choi, Nicole. The architecture of today's LLM applications. *GitHub*. [Online] https://github.blog/2023-10-30-the-architecture-of-todays-llm-applications/.

7. Valenzuela, Andrea. LLM Classification: How to Select the Best LLM for Your Application. *Datacamp*. [Online] https://www.datacamp.com/tutorial/llm-classification-how-to-select-the-best-llm-for-your-application.

8. Poro 34B Model Card. *HuggingFace*. [Online] https://huggingface.co/LumiOpen/Poro-34B.

9. Huang, Xian. An (Opinionated) Checklist to Choose a Vector Database. *Pinecone*. [Online] https://www.pinecone.io/learn/an-opinionated-checklist-to-choose-a-vector-database/.

10. Ali, Mutahar. Optimizing RAG: A Guide to Choosing the Right Vector Database. *Medium*. [Online] https://medium.com/@mutahar789/optimizing-rag-a-guide-to-choosing-the-right-vector-database-480f71a33139.

11. Joshi, Apoorva. RAG Series Part 1: How to Choose the Right Embedding Model for Your Application. *MongoDB*. [Online] https://www.mongodb.com/developer/products/atlas/choose-embedding-model-rag/.

12. Otten, Neri Van. What is LLM Orchestration? *Spot Intelligence*. [Online] https://spotintelligence.com/2023/11/17/llm-orchestration-frameworks/.

13. Padhy, Dr Rabi Prasad. Exploring Popular LLM Orchestration Frameworks. *LinkedIn*. [Online] https://www.linkedin.com/pulse/exploring-popular-llm-orchestration-frameworks-dr-rabi-prasad-hjxkc/.

14. Vartak, Manasi. Six Risks Of Generative AI. *Forbes*. [Online] https://www.forbes.com/sites/forbestechcouncil/2023/06/29/six-risks-of-generative-ai/.

15. S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, Amitava Das. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. [Online] 2024. https://arxiv.org/abs/2401.01313.

16. What are AI hallucinations? *IBM*. [Online] https://www.ibm.com/topics/ai-hallucinations.

17. Arsanjani, Ali. Navigating the Challenges of Hallucinations in LLM Applications: Strategies and Techniques for Enhanced Accuracy. *Medium*. [Online] https://dr-arsanjani.medium.com/navigating-the-challenges-of-hallucinations-in-llm-applications-strategies-and-techniques-for-ab2b5ddc4a63.

18. Marr, Bernard. Generative AI And Data Protection: What Are The Biggest Risks For Employers? *Forbes*. [Online] https://www.forbes.com/sites/bernardmarr/2024/05/27/generative-ai-and-data-protection-what-are-the-biggest-risks-for-employers/.

19. Baig, Anas. Navigating Generative AI Privacy : Challenges & Safeguarding Tips. *Securiti*. [Online] https://securiti.ai/generative-ai-privacy/.

20. The EU Artificial Intelligence Act. Up-to-date developments and analyses of the EU AI Act. [Online] https://artificialintelligenceact.eu/.

21. Emilio, Ferrara. Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. [Online] 2023. https://arxiv.org/pdf/2304.07683.

22. Guvvala, Sujitha. Bias Mitigation in Generative AI. *Analytics Vidhya*. [Online] https://www.analyticsvidhya.com/blog/2023/09/bias-mitigation-in-generative-ai/.

23. Falck, Juha. 5 steps to getting your organization started with generative AI. *Qvik*. [Online] https://qvik.com/news/5-steps-to-getting-your-organization-started-with-generative-ai/.