MCKNIGHT CONSULTING GROUP

From Pre-Trained to Fine-Tuned: How to Get the Most Out of Vector, RAG, and Small Language Models

Presented by: William McKnight

"#1 Global Influencer in Big Data" Thinkers360

President, McKnight Consulting Group

3 X Inc 5000

/in/wmcknight

www.mcknightcg.com
(214) 514-1444

McKnight Consulting Group Partial Technology Implementation Expertise

Big/Analytic/Vector/Mixed Data Management

Data Movement and APIs

Data Management

Operational/Transactional Data Management

MCKNIGHT CONSULTING GROUP

Dataversity Analytics Architecture with William McKnight 2026

1. 2026 Trends in Analytic Architectures
2. What Does Information Management Maturity Look Like in 2026
3. The Data Product Revolution: Unlocking Business Value
4. Building Effective RAG Applications
5. Data Professionals in the AI Age: What's Next?
6. The Master Data Management Dilemma: To Buy or Build, That is the Question: Benchmark Completed
7. Promising AI Use Cases for the Enterprise in 2026
8. Data Mastery: William Answers Your Questions
9. Data Pipeline Engineering Strategies
10. How to Work with Open Table Formats
11. The ROI of Agentic AI: Strategies for Success
12. Data Architecture 2027: What enterprises are building today and why

AI Interest is at an all-time High and Growing

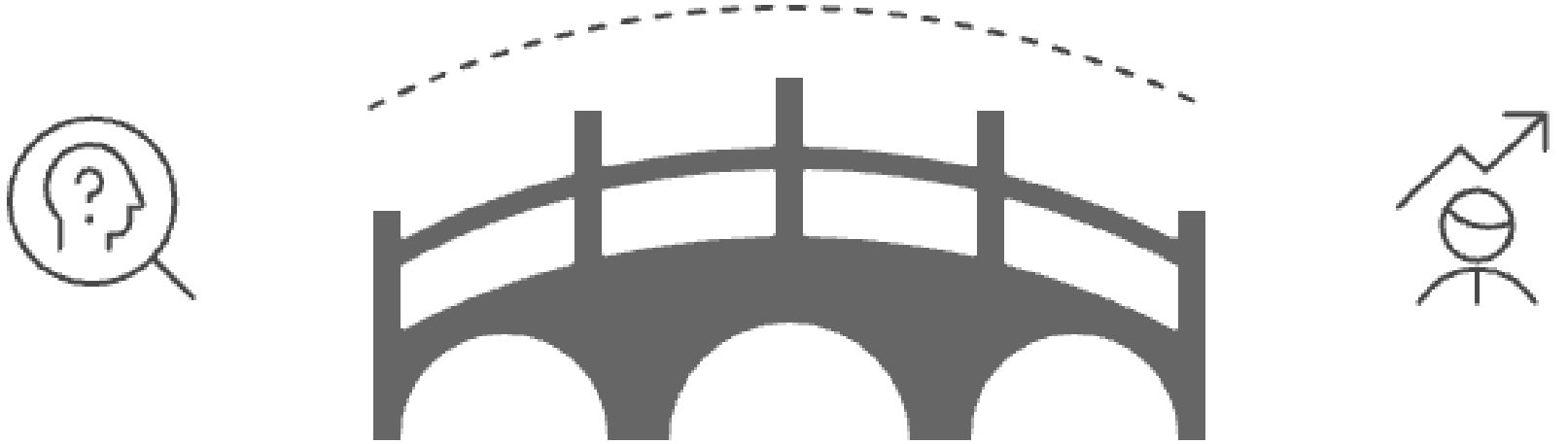
Hundreds of companies will be built around an API for something like ChatGPT

Startups will not be able to create the AI themselves, but they can use the APIs

Nearly every industry and nearly every vertical is being transformed today

Companies are using these techniques in software and statistical models to make predictions and drive businesses forward in a way that they're not able to with only humans

AI: Bridging Hype to Real-World Value



AI Hype

Unrealistic expectations and implementation challenges

AI Value

Tangible benefits across industries

Outcomes of Using AI Well

- **Increased Efficiency:**
  AI agents can process large amounts of data, freeing up human resources for more strategic and creative tasks.

- **Improved Customer Experience:**
  AI-powered customer service agents can resolve issues faster and more accurately, leading to higher customer satisfaction.

- **Enhanced Decision-Making:**
  AI agents can analyze complex data sets, providing valuable insights for informed decision-making.

- **Cost Savings:**
  By automating repetitive tasks and optimizing resources, AI agents can help businesses reduce operational expenses.

- **Data-Driven Insights:**
  AI agents can analyze large datasets, providing businesses with actionable insights to drive growth and innovation.

AI Considerations

Not necessarily unique to AI….

AI agents make **all enterprise data accessible** (audio, video, text, and alpha-numeric).

High-velocity, fast-paced streaming data is crucial as a "foundation for artificial intelligence." This real-time data flow supports **continuous learning, automation, and fine-tuning**.

The expectation is that **unstructured data will be "at parody with structured data."**

Get the data act together – is the "lifeblood of AI agents".

- Contextual relevance
- Temporal relevance
- Spatial relevance
- Causality reference
- Real-time data
- Multi-modal data

Large Language Models (LLMs)

History of LLMs

- Evolution of neural networks
- RNNs predicted next word in sentence in early 2000s
- 2017 Google DeepMind Team paper on Transformers
- 2018 Open AI developed GPT-1
- Traditional programming is instruction-based
- LLMs is teaching not ‘how’ but giving examples and asking it to learn

Large Language Models

A type of artificial intelligence (AI) model designed to process and understand human language

- Trained on vast amounts of text data
- Learns patterns and relationships in language

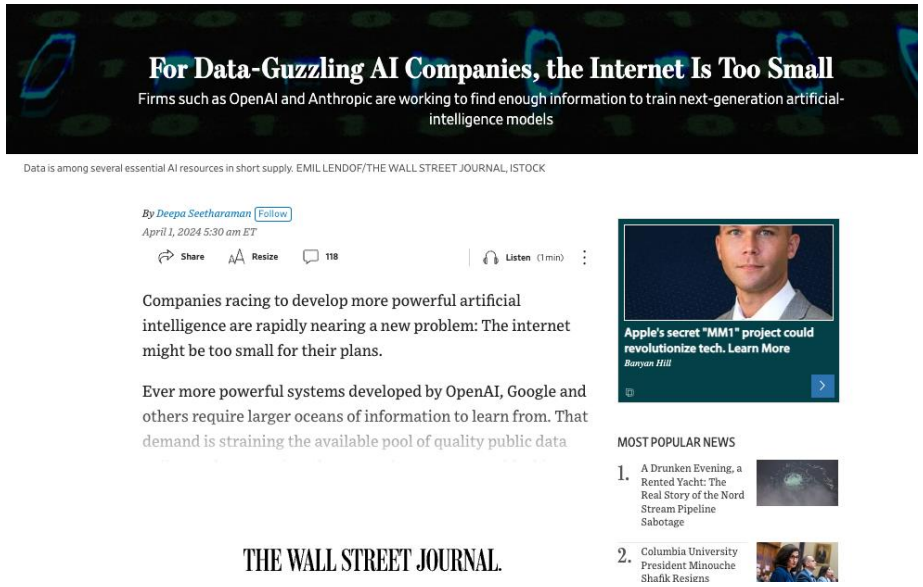
- **Architecture**:
  - Deep learning architectures
  - Transformers
  - Recurrent neural networks (RNNs)
  - Convolutional neural networks (CNNs)
- **Capabilities**:
  - Language understanding
  - Text generation
  - Translation
  - Summarization
  - Sentiment analysis
  - Question answering

LLMs are Trained

Large Language Models (LLMs) are trained on vast amounts of text data to learn patterns and relationships in language. This training process enables LLMs to understand and generate human-like language.

Training Data:

1. **Web pages**
2. **Books and articles**
3. **User-generated content** (e.g., social media, forums)
4. **Product reviews**
5. **Wikipedia**
6. **Synthetic data**



For Data-Guzzling AI Companies, the Internet Is Too Small

Firms such as OpenAI and Anthropic are working to find enough information to train next-generation artificial-intelligence models

Data is among several essential AI resources in short supply. EMIL LENDOF/THE WALL STREET JOURNAL, ISTOCK

By Deepa Seetharaman Follow

April 1, 2024 5:30 am ET

Share Resize 118 Listen (1 min)

Companies racing to develop more powerful artificial intelligence are rapidly nearing a new problem: The internet might be too small for their plans.

Ever more powerful systems developed by OpenAI, Google and others require larger oceans of information to learn from. That demand is straining the available pool of quality public data

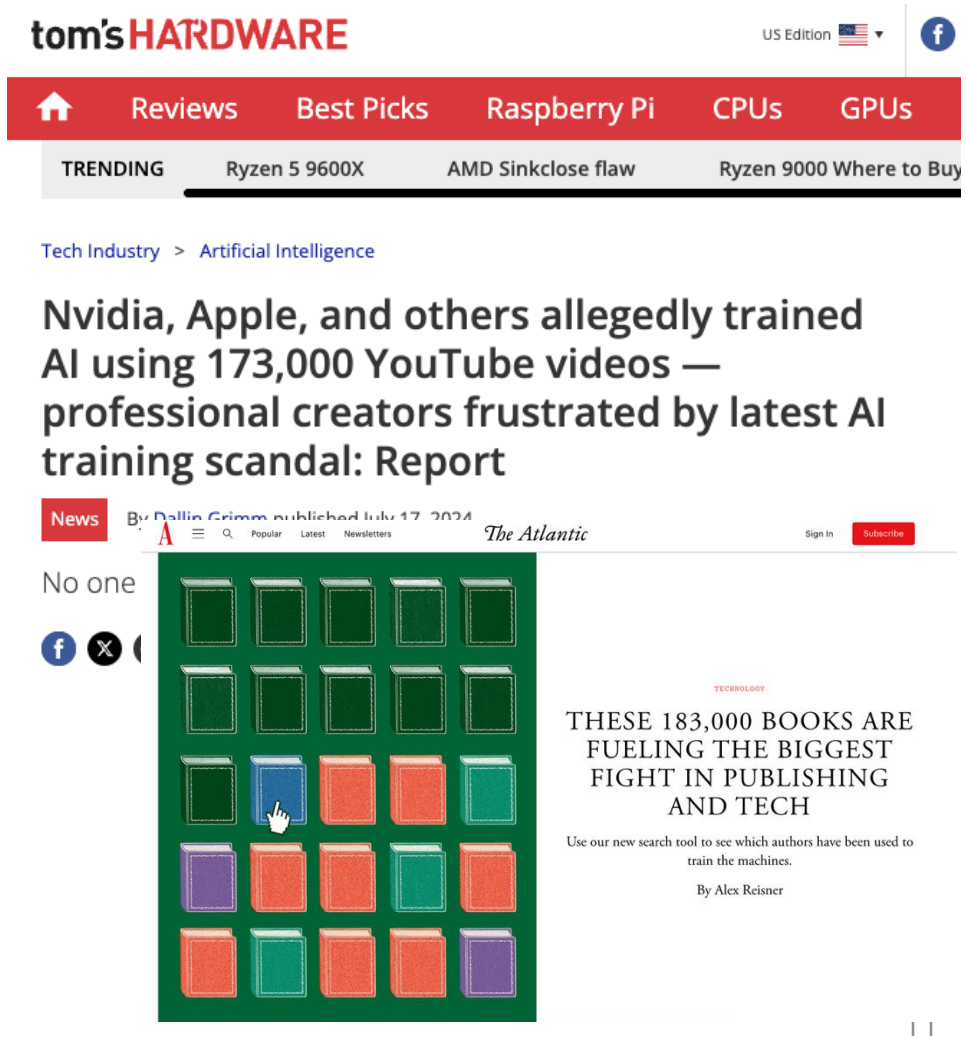
Apple's secret "MM1" project could revolutionize tech. Learn More

Banyan Hill

MOST POPULAR NEWS

1. A Drunken Evening, a Rented Yacht: The Real Story of the Nord Stream Pipeline Sabotage
2. Columbia University President Minouche Shafik Resigns

THE WALL STREET JOURNAL.



tom's HARDWARE

US Edition

Reviews Best Picks Raspberry Pi CPUs GPUs

TRENDING Ryzen 5 9600X AMD Sinkclose flaw Ryzen 9000 Where to Buy

Tech Industry > Artificial Intelligence

Nvidia, Apple, and others allegedly trained AI using 173,000 YouTube videos — professional creators frustrated by latest AI training scandal: Report

News By Dallin Grimm published July 17, 2024

No one

The Atlantic

Popular Latest Newsletters Sign In Subscribe

TECHNOLOGY

THESE 183,000 BOOKS ARE FUELING THE BIGGEST FIGHT IN PUBLISHING AND TECH

Use our new search tool to see which authors have been used to train the machines.

By Alex Reisner

EU AI Act

- Regulated AI
- Common risk structure for AI
- Specific requirements for GenAI providers and users
  - Providers include customers passing along access to LLMs
- Fines
- Began 8/1/24
- It's about change management
- Companies in US that operate in EU need to comply

LLM Steps

1. Tokenization
2. Embeddings (vectors)
3. Transformers

How LLMs are Trained

1. **Data Collection**: Gathering a massive dataset of text from various sources (e.g., books, articles, websites).
2. **Preprocessing**: Cleaning and formatting the data (e.g., tokenization, stopword removal).
3. **Model Architecture**: Designing the LLM's architecture (e.g., transformer, recurrent neural network).
4. **Training**: Feeding the preprocessed data into the model, adjusting parameters to minimize errors.

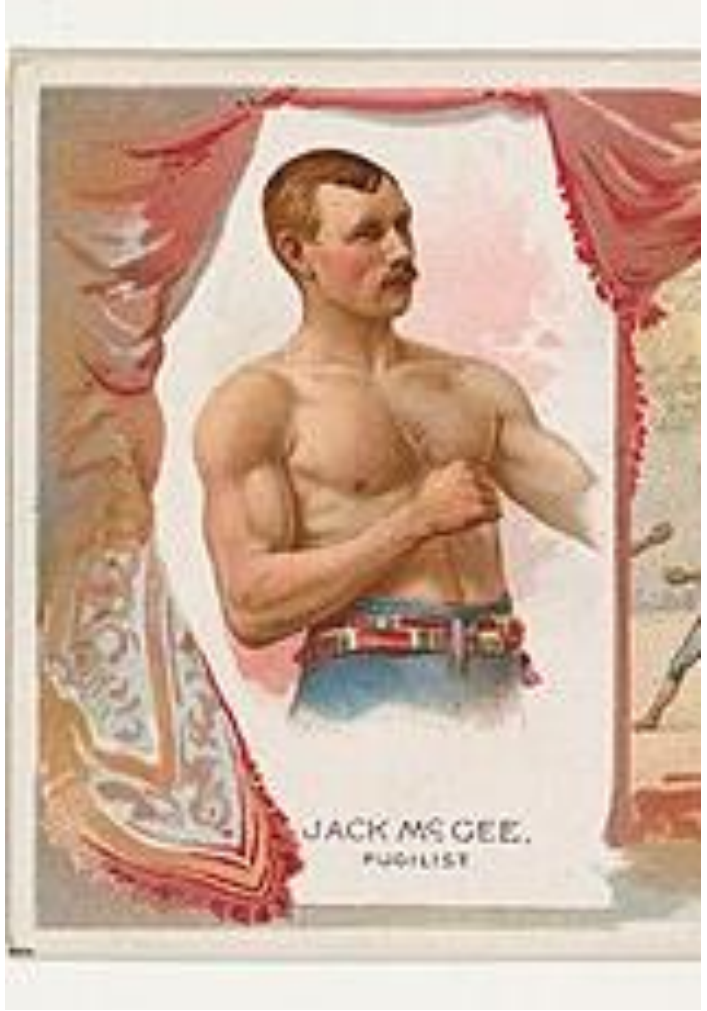
To Build an LLM

Building an LLM requires significant expertise, resources, and effort.

1. **Data**
2. **Compute Resources**
3. **Model Architecture**
4. **Training: Algorithm, Objective, Batching**
5. **Software and Tools:** TensorFlow, PyTorch, or JAX and libraries
6. **Expertise**

Challenges

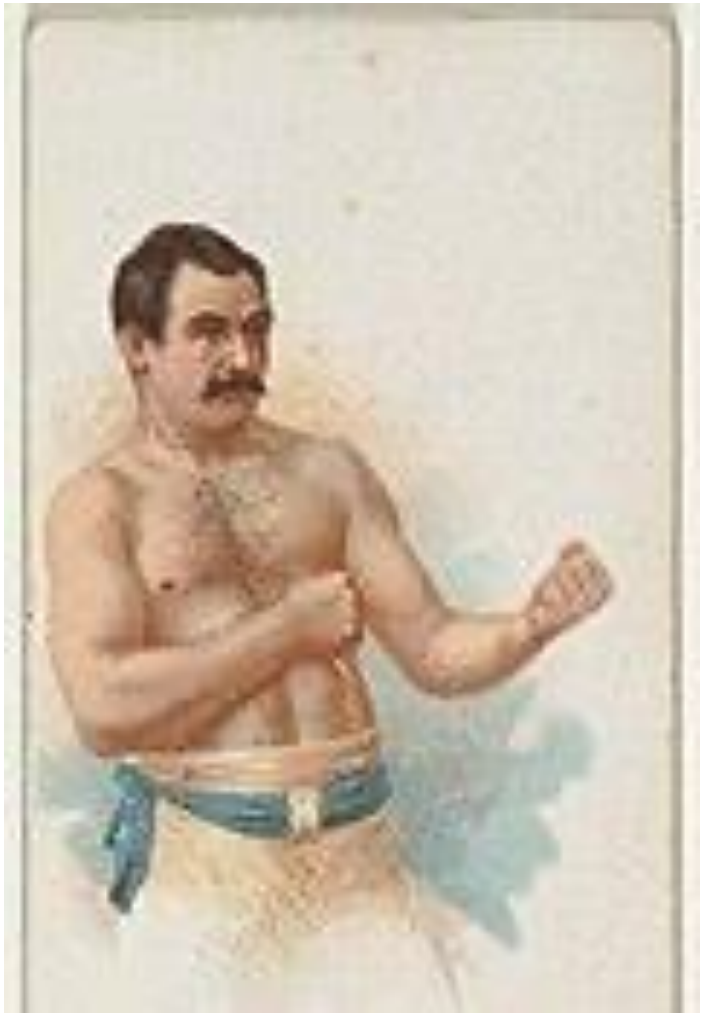
- **Data quality and availability**
- **Computational resources and scalability**
- **Model complexity and training instability**
- **Evaluation and fine-tuning**



Who Has Built LLMs

IBM

LLM Use Cases



- Customer Service
- Advisory
- Programming Assistants
- Summarization
- Language Translation
- Essay Writing
- Image Captioning

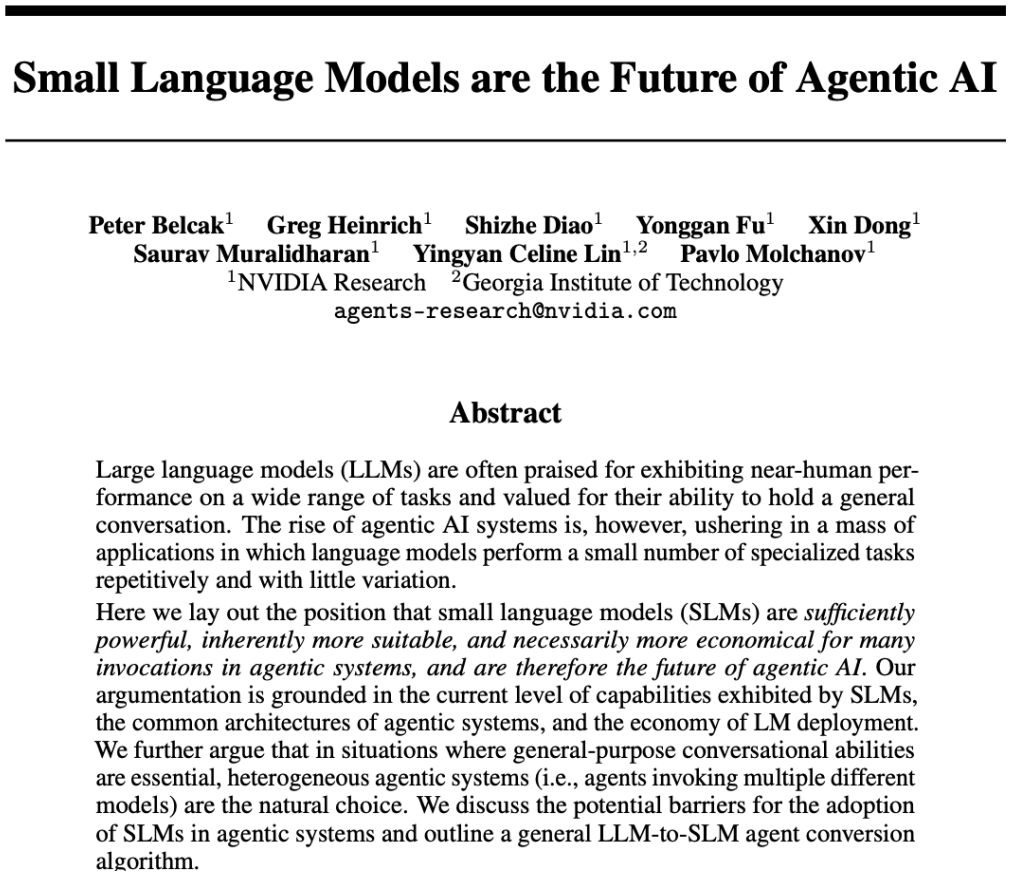
How to use LLMs

- Cloud
- License
- API
  - Ie, Meta AI's official API for LLaMA
  - Register for an API Key
  - Configure API Settings
  - Use Python/other to send API requests to LLaMA
  - Receive output data
- Pipelines

```
import requests

api_key = "YOUR_API_KEY"
model_name = "llama"
input_text = "Your input text here"

url = f"https://api.huggingface.co/models/{model_name}"
headers = {"Authorization": f"Bearer {api_key}"}
data = {"inputs": input_text, "parameters": {"temperature": 0.7}}

response = requests.post(url, headers=headers, json=data)
output = response.json()["generated_text"]

print(output)
```

Small Language Models: The Real AI Revolution

- Efficiency: Less computational resources and energy required.
- Faster deployment: Quicker response times and real-time processing.
- Lower costs: Less expensive to train and maintain.
- Specialized knowledge: Can be fine-tuned for specific domains or tasks.
- Easier interpretability: Better understanding of decision-making processes.
- Practical applications: Suitable for applications like chatbots, virtual assistants, and language translation.



Small Language Models are the Future of Agentic AI

Peter Belcak[1] **Greg Heinrich**[1] **Shizhe Diao**[1] **Yonggan Fu**[1] **Xin Dong**[1]
Saurav Muralidharan[1] **Yingyan Celine Lin**[1,2] **Pavlo Molchanov**[1]
[1]NVIDIA Research [2]Georgia Institute of Technology
agents-research@nvidia.com

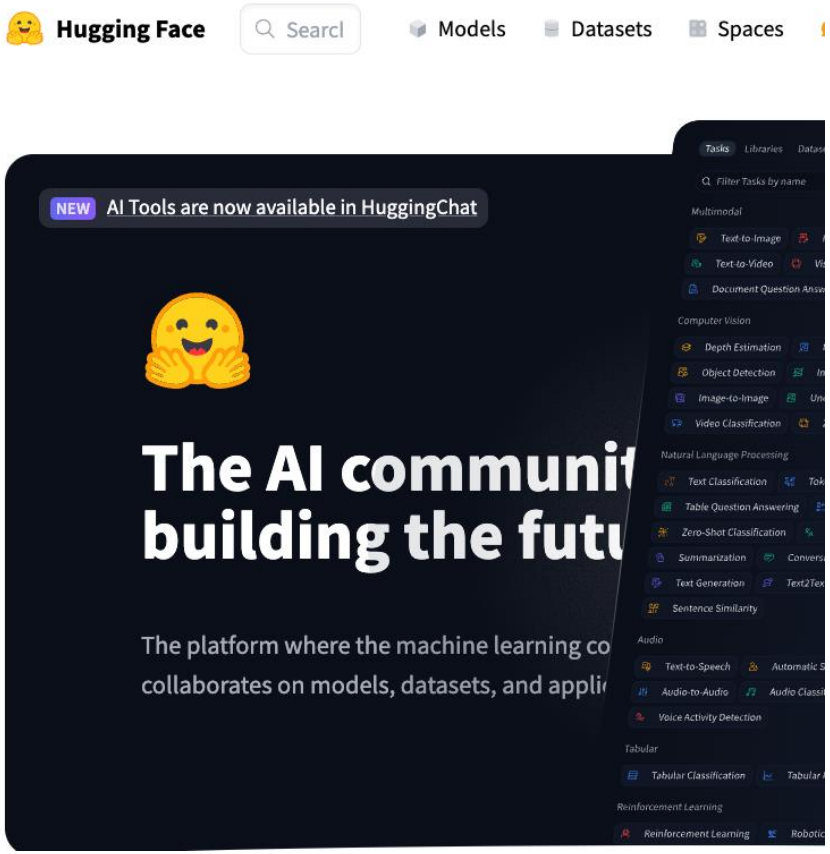
Abstract

Large language models (LLMs) are often praised for exhibiting near-human performance on a wide range of tasks and valued for their ability to hold a general conversation. The rise of agentic AI systems is, however, ushering in a mass of applications in which language models perform a small number of specialized tasks repetitively and with little variation.

Here we lay out the position that small language models (SLMs) are *sufficiently powerful, inherently more suitable, and necessarily more economical for many invocations in agentic systems, and are therefore the future of agentic AI*. Our argumentation is grounded in the current level of capabilities exhibited by SLMs, the common architectures of agentic systems, and the economy of LM deployment. We further argue that in situations where general-purpose conversational abilities are essential, heterogeneous agentic systems (i.e., agents invoking multiple different models) are the natural choice. We discuss the potential barriers for the adoption of SLMs in agentic systems and outline a general LLM-to-SLM agent conversion algorithm.

Hugging Face

- Hugging Face is best known for its wide range of pre-trained models and a simple interface for using them.
- Hugging Face offers a vast collection of pre-trained models for various NLP tasks, such as language translation, question answering, and text generation.
- Hugging Face has a large and active community of developers, researchers, and practitioners who contribute to the library and share their knowledge.
- Hugging Face's library provides an interface for integrating pre-trained models into applications.



Future of Language Models

- Fact-Checking Itself
- Multi-Modality
- Improve Reasoning Ability
- Bigger Context Windows
- Small Language Models

Pretraining and Fine Tuning Models

Fine Tuning Models: The Art of Biasing Intelligence

- Fine-tuning and customization of models, particularly Large Language Models (LLMs), are crucial processes detailed in the sources, especially regarding the operationalization and specialization of AI solutions.
- General-purpose LLMs often require fine-tuning to meet the specific needs of various industries, such as the legal, finance, or healthcare sectors. The main goals of customization are to ensure that models:
  - Understand domain-specific terminology.
  - Align with compliance regulations and ethical considerations.
  - Meet business objectives and optimize performance while maintaining the highest standards of accuracy and relevance.
- Fine-tuning and customizing LLMs are described as critical steps for developing domain-specific, high-performance AI models. This practice will remain a key enabler of industry-specific, reliable, and ethical AI solutions.

Fine Tuning Techniques

- Supervised Fine-Tuning: This involves training a pre-trained model on a labeled dataset to align it with specific use cases.
- Reinforcement Learning from Human Feedback (RLHF): This method enhances model performance by incorporating human preferences.
- Parameter-Efficient Fine-Tuning (e.g., LoRA): LoRA works by introducing trainable layers that adapt pre-trained models to new tasks, while freezing most of the model's weights, minimizing resource consumption.
- Prompt Engineering and Few-Shot Learning are also techniques for customizing LLMs.
- Self-supervised learning (SSL) models, such as GPT-4 and BERT, learn contextual relationships that can be fine-tuned for downstream tasks.

Vector Databases and RAG

Enter Vector Databases

- Conventional databases lack the structural capability to accommodate imprecise comparative inquiries such as "which items are comparable to this one?"
- The exploration of machine representations of datasets such as text, voice, image, and molecular structures is underway as ML and LLMs are applied to novel problems.
- Vector emerged to address new issues, much like the NoSQL generation of databases did.
- User inquiries evolved in tandem with the influx of machine representations of data.
- To address them, vector databases, a novel technology, were required.

Imprecise/Similarity Search

- Ever searched for something with unclear detail?
- Struggling to search because you lack the exact keywords?
- Remembering actors but forgetting the movie title?
- Frustrated with outdated information in search results?
- Frustrated by generalized Large Language Models (LLMs)?
- If any of these sound familiar, then vector search can be your solution

Vector Search

- Search based on meaning, not just keywords
- Leverages machine learning models called encoders for powerful results
- Embeddings
  - Text, audio, images transformed into a numeric string called "vectors"
  - High-dimensional arrays with semantic meaning
  - Makes data available to AI
- Benefits of Vector Search
  - Semantic Understanding
  - Scalable
  - Flexible – anything can be vectorized

- **Example: [-0.0385810, -0.1348581, 0.0184810, -0.138542, -0.1984815, 0. 12498134, 0.0124897, -0.021858, -0.0002384, -0.024911, 0.199248284, ...]**

Vectors

- Numeric representations
- "The only thing we have to fear is fear itself" = [-2.345, 7.812, -1.009, 4.567, 0.123, 9.998,…]
- "To be or not to be, that is the question" = [-3.141, 2.718, -8.000, 1.618, 4.200, 7.539, …]
- "I think, therefore I am" = [-5.827, 0.123, 9.000, -1.414, 3.142, 6.789, …]

Vector Embeddings come from an Embedding Model

- Example Embedding Models are provided by OpenAI, Cohere, Google Vertex, HuggingFace

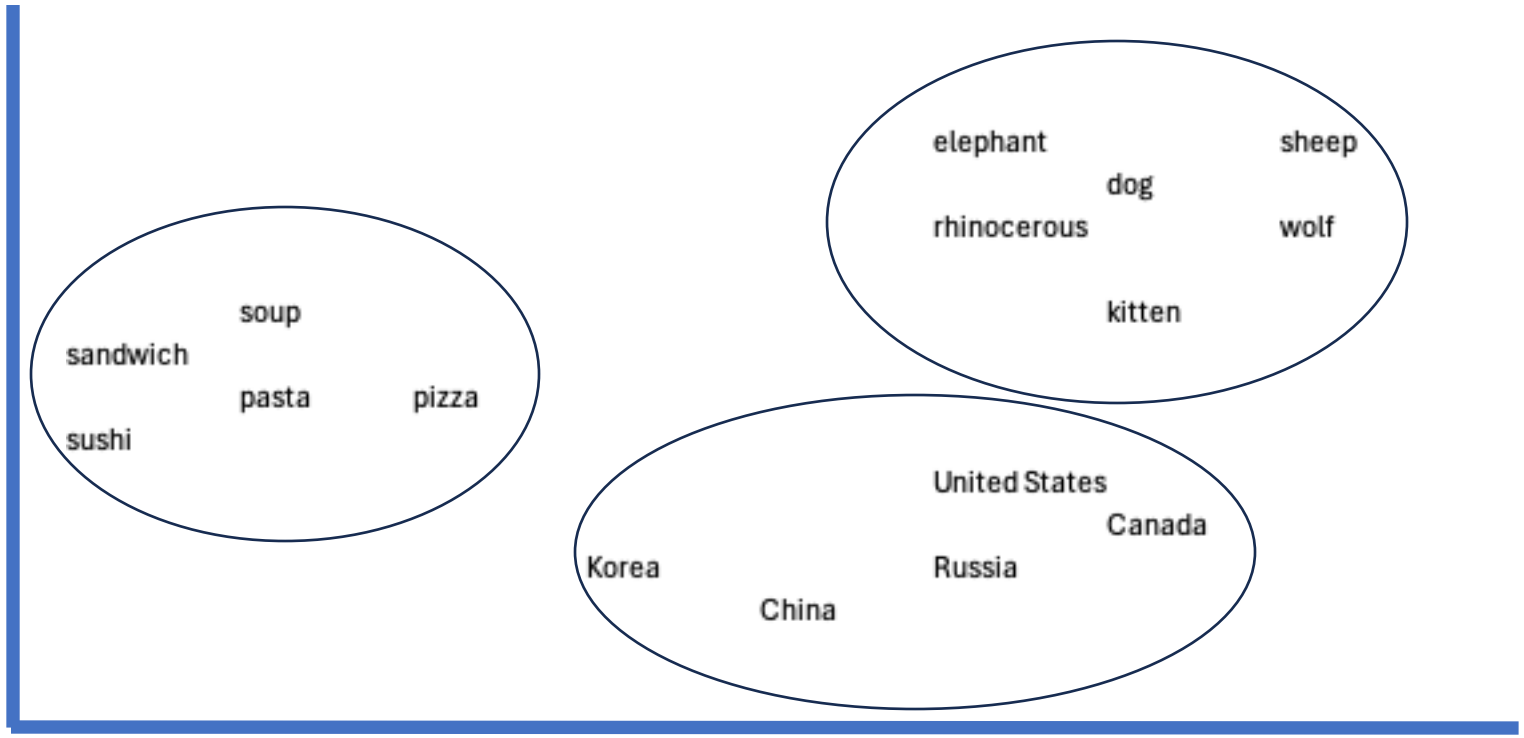
OpenAI

Hugging Face

cohere

[-2.345, 7.812, -1.009, 4.567, 0.123, 9.998,…]
[-3.141, 2.718, -8.000, 1.618, 4.200, 7.539, …]
[-5.827, 0.123, 9.000, -1.414, 3.142, 6.789, …]

Data Clustering in Space



Word2vec for text

- Word2Vec is a technique for representing words as numerical vectors.
- These vectors capture the semantic meaning and relationships between words.
- Word2Vec models are trained on massive datasets of text.
- The model analyzes the context in which words appear, learning their meaning based on surrounding words.
- Words with similar meanings have similar vector representations in the embedding space.
  - Imagine "king" and "queen" having vectors close together, while "king" and "car" would be farther apart.
- Word2Vec embeddings fuel various applications:
  - Recommendation systems suggesting similar products based on user searches.
  - Machine translation finding the closest meaning in another language.
  - Chatbots understanding the intent behind user queries.

DBMS are adding Vector Support

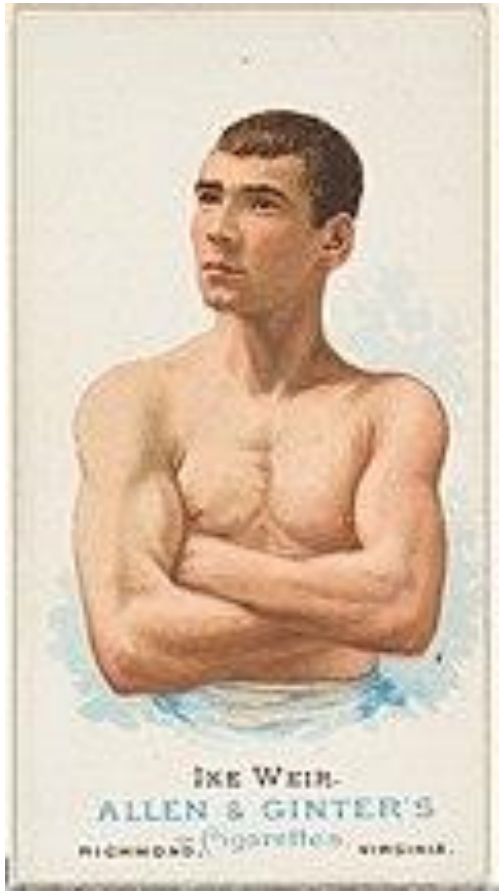
- Vector data has a wide range of use cases, ultimately limited only by creativity and imagination.
- Isolated data sources (silos) can limit the accuracy and efficiency of analytics, including those involving vectors.
- Developers are overwhelmed by managing an abundance of individual tools and interfaces.
- Both developers and enterprises prioritize consolidation of their technology stacks for better manageability.
- The ideal scenario is a single interface offering broad capabilities across various data model problems, without sacrificing functionality.

Vector Search

- By using the distance between high dimensional vectors, vector search enables us to search through data and find relevant results.
- When calculating the distance between your query and stored vectors at query time, it uses machine learning models to embed the data.

K-Nearest Neighbor

- Closest thing to exact search for vectors; it finds the perfect nearest neighbors.
- Typical approach to classification problems is k-nearest neighbors (KNN) Classification.
- KNN predicts the label (class) or value (regression) for a new data point by looking at its k nearest neighbors in the training data.
- Contrary to the notion of "exact search," KNN doesn't necessarily find perfect matches in the training data.
- It focuses on identifying the k data points most similar (closest) to the new point based on a chosen distance metric (e.g., Euclidean distance).



Hierarchical, Navigable Small World

- HNSW utilizes a graph-like structure with layers, enabling efficient traversal to find approximate nearest neighbors.
- Hierarchical Navigable Small World (HNSW) is an indexing strategy for Approximate Nearest Neighbor (ANN) search.
- HNSW enables fast retrieval of "mostly" nearest neighbors for K-Nearest Neighbors (KNN), improving efficiency for large datasets.
- HNSW utilizes a graph-like structure with layers, where similar data points are connected within layers.
- During a search, the algorithm traverses the layers of the graph, efficiently navigating towards potential nearest neighbors.
- HNSW offers a trade-off between finding the absolute closest neighbors (accuracy) and achieving high search speed.
- It is ideal for real-time applications with large datasets where KNN needs a performance boost.

Similarity Functions

- Eucliden
  - Measures the straight-line distance between two points in a multidimensional space.
  - Commonly used for numerical data with real number values. Larger distance indicates less similarity.
- Cosine
  - Measures the directional similarity between two vectors. Useful for data where the magnitude (length) of the vectors may not be important.
  - Values range from -1 (completely opposite) to 1 (identical).
- Dot Product
  - Calculates the product of corresponding components of two vectors and then sums them up.
  - Closely related to cosine similarity, but it considers the magnitudes of the vectors as well.
  - Larger dot product indicates greater similarity.

RAG (Retrieval-Augmented Generation)

This is a technique that combines language models with vector databases. The process:

1. The LM receives a prompt or question.
2. The RAG system uses the vector database to find similar information related to the prompt.
3. The LM uses the retrieved information to improve its understanding of the context and generate a more accurate and relevant response.

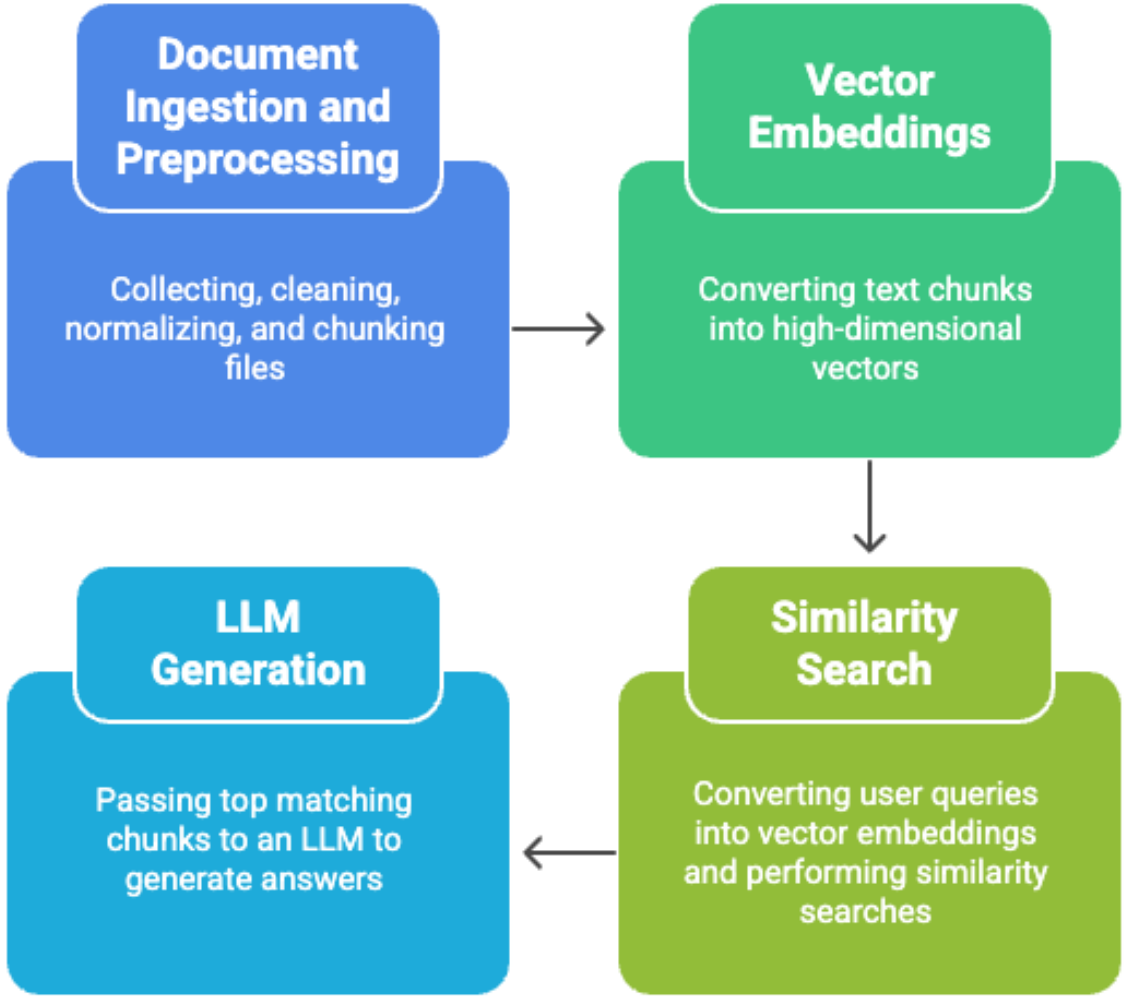
RAG and Language Models

Retrieval-Augmented Generation combines the strengths of LMs and external knowledge sources

Enhances accuracy, relevance, and diversity of generated content

- **LMs and RAG: A Powerful Duo**:
  - LMs provide context and language understanding
  - RAG leverages external knowledge to augment LLM capabilities
- **Key Benefits**:
  - Improved accuracy and relevance
  - Increased diversity and novelty
  - Enhanced ability to handle nuanced and complex topics
- **Applications**:
  - Content generation (text, chatbots, etc.)
  - Question answering and information retrieval
  - Summarization and knowledge graph construction

RAG Application Process

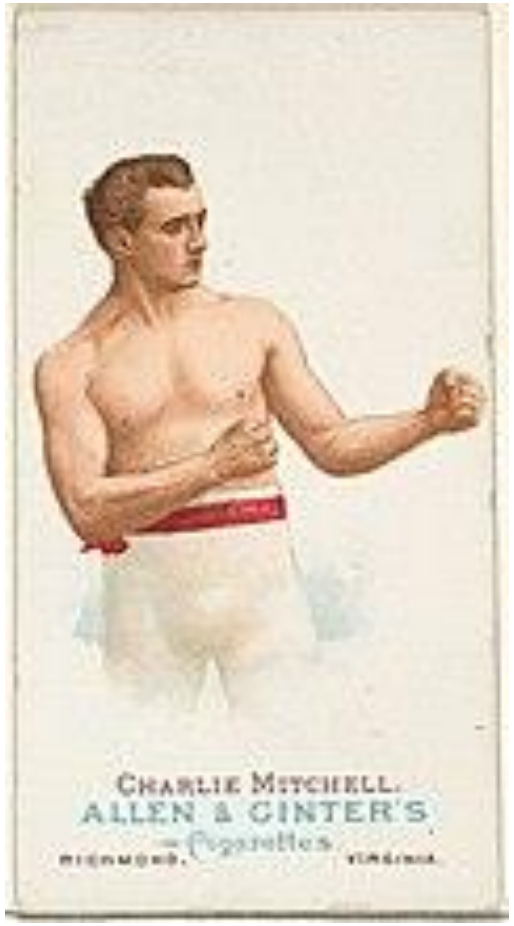


Example RAG Stack

Database + AI	Amazon RDS for PostgreSQL
	Amazon KMS
	Amazon EC2 (Embedding)
Distributed HA	Amazon EC2 (Embedding)
Message Queue	Amazon MSK
Data Lake	Amazon MSK
Observability & Monitoring	Amazon CloudWatch
Security & Compliance	Amazon IAM

Effort, Complexity, and Total Cost of Ownership

DIY	Development Story Points (one-time)	Production Story Points (per year)
Source Document Ingest	13	78
Document Preprocessing and Chunking	13	52
Embedding Microservice	21	52
LLM Generation Service	63	78
Retriever Service	39	52
Logs	32	52
Metrics, Dashboards, and Alarms	13	26
Access Control, Secrets & Encryption	5	26
DIY Total	**199**	**416**



Simple Vector_Distance Function

SELECT…

FROM house_for_sale

WHERE price <= (SELECT budget FROM customer …)

AND city in (SELECT search_city FROM customer …)

ORDER BY vector_distance(house_vectors, :input_vector);

Applications of Vector Search in Company

- Copilots
- Customer service
- IT support
- Employee onboarding
- Chip design
- Drug discovery
- Fraud investigation
- Screening resumes
- Answering common employee questions
- Curriculum design
- Demand planning for supplies
- RFP Automation
- Document and research summarization
- Patent drafting
- Chatbot for support
- Ticket classification
- Logistics optimization
- Supplier Analysis

AI Concepts Companion

Retrieval-Augmented Generation (RAG)

RAG combines pre-trained language models with external knowledge retrieval. It dynamically fetches relevant information from databases or documents to enhance response accuracy and provide up-to-date information.

Data Access
External, Real-time

Memory
Unlimited (via retrieval)

Update Frequency
Real-time possible

Computational Cost
Higher per query

Small Language Models (SLMs)

SLMs are compact language models with fewer parameters, designed for efficiency and specific tasks. They offer faster inference, lower computational requirements, and can be deployed on edge devices while maintaining reasonable performance.

Model Size
1B-7B parameters

Deployment
Edge, Mobile, Local

Speed
Fast inference

Specialization
Domain-specific

Pre-training

Pre-training is the initial phase where models learn general language understanding from massive datasets. This foundational training creates versatile models that understand syntax, semantics, and world knowledge.

Data Scale
Billions of tokens

Learning Type
Unsupervised

Duration
Weeks to months

Purpose
General knowledge

Fine-tuning

Fine-tuning adapts pre-trained models for specific tasks or domains. It involves continued training on smaller, task-specific datasets to optimize performance for particular use cases while preserving general knowledge.

Data Scale
Thousands to millions

Learning Type
Supervised

Duration
Hours to days

Purpose
Task specialization

Use RAG When...; Use Small Language Models When...

- Use RAG When...
  - Business Intelligence: Building dashboards that need real-time data analysis and reporting
  - Knowledge Base Q&A: Customer support systems accessing constantly updated documentation
  - Medical Diagnosis: Systems that need access to latest research papers and treatment protocols
  - Legal Research: Applications requiring access to current laws, cases, and regulations
  - Financial Analysis: Trading systems needing real-time market data and news analysis
  - Scientific Research: Tools that must reference the latest published studies and data
  - Best For: Dynamic information needs, factual accuracy, real-time updates
- Use Small Language Models When...
  - Mobile Apps: On-device text processing, autocomplete, and writing assistance
  - Edge Computing: IoT devices requiring local language processing capabilities
  - Privacy-Critical: Healthcare or financial apps needing on-premise processing
  - Real-time Systems: Gaming NPCs, chatbots with strict latency requirements
  - Cost-Sensitive: Startups or applications with tight computational budgets
  - Offline Applications: Systems that must work without internet connectivity
  - Best For: Resource constraints, speed requirements, privacy needs

Use Pre-training When...; Use Fine-tuning When...

- Use Pre-training When...
  - New Languages: Creating models for under-resourced languages or dialects
  - Novel Domains: Scientific fields with specialized vocabularies (chemistry, physics)
  - Enterprise Foundation: Companies building proprietary models from scratch
  - Research Projects: Academic work exploring new architectures or techniques
  - Massive Datasets: Applications with unique, large-scale text corpora
  - Specialized Tasks: Applications requiring fundamentally different language understanding
  - Best For: New languages, massive resources, research purposes
- Use Fine-tuning When...
  - Content Generation: Brand-specific writing styles, marketing copy, technical documentation
  - Classification Tasks: Sentiment analysis, document categorization, content moderation
  - Translation: Domain-specific translation (medical, legal, technical)
  - Conversational AI: Customer service bots with specific personality traits
  - Information Extraction: Named entity recognition, relationship extraction
  - Summarization: News articles, research papers
  - Best For: Task specialization, performance optimization, adapting new tasks

Summary

- AI is bridging its hype to real-world value
- Large language models are trained by exposing neural networks to vast amounts of text data and adjusting their internal weights to predict
- In Retrieval-Augmented Generation (RAG), a vector database uses algorithms like HNSW and kNN for similarity search to efficiently retrieve the most relevant document embeddings
- Small language models are lightweight versions of large language models designed to run efficiently on limited hardware
- Pretraining teaches a model general knowledge, and fine-tuning tailors it to a specific task

MCKNIGHT CONSULTING GROUP

From Pre-Trained to Fine-Tuned: How to Get the Most Out of Vector, RAG, and Small Language Models

Presented by: William McKnight

"#1 Global Influencer in Big Data" Thinkers360

President, McKnight Consulting Group

3 X Inc 5000

/in/wmcknight

www.mcknightcg.com
(214) 514-1444