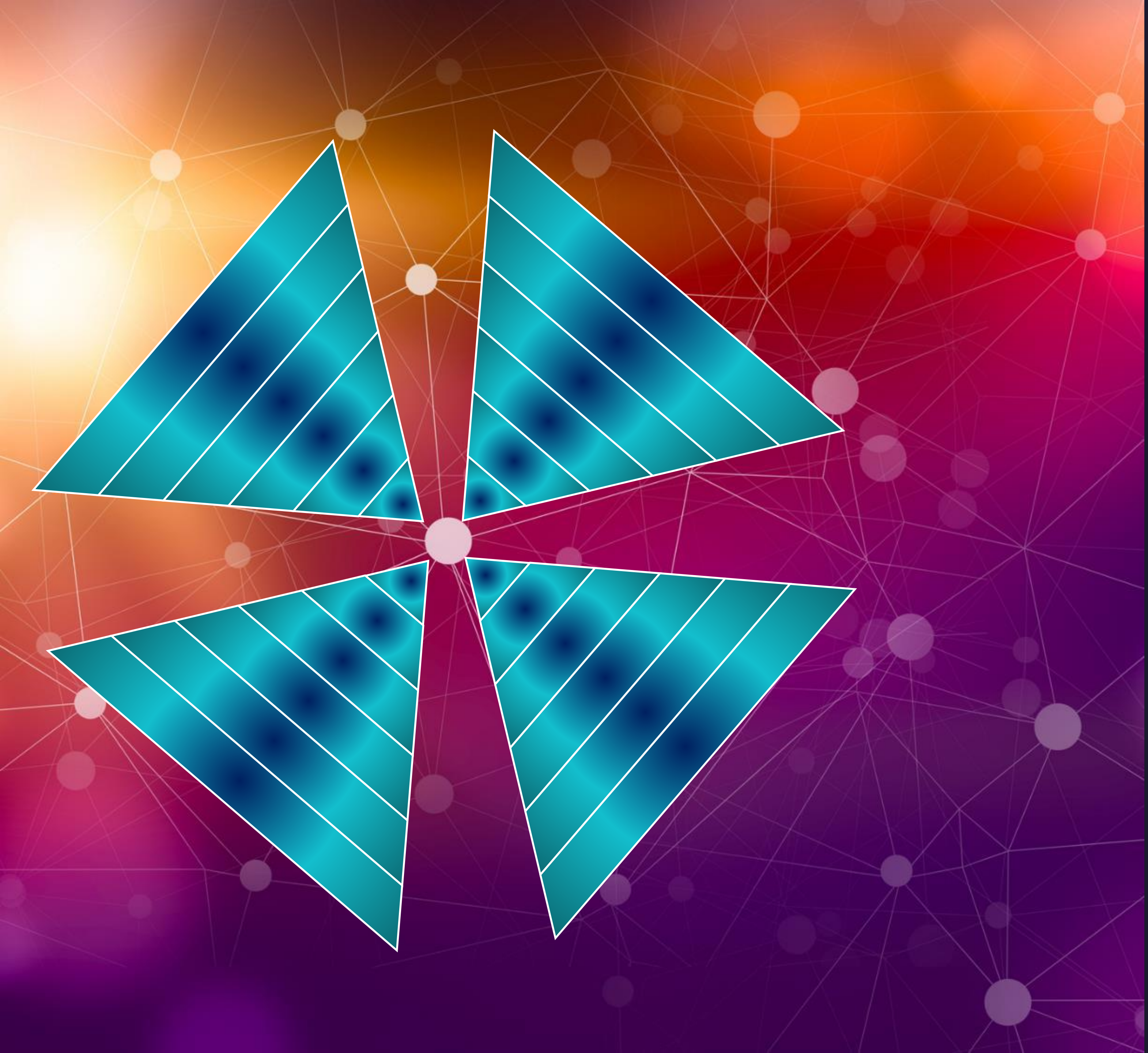


Practical ChatGPT

and other large language models

Michael Iantosca
Senior Director of Content Platforms
Michael.Iantosca@avalara.com

Avalara

https://thinkingdocumentation.com/

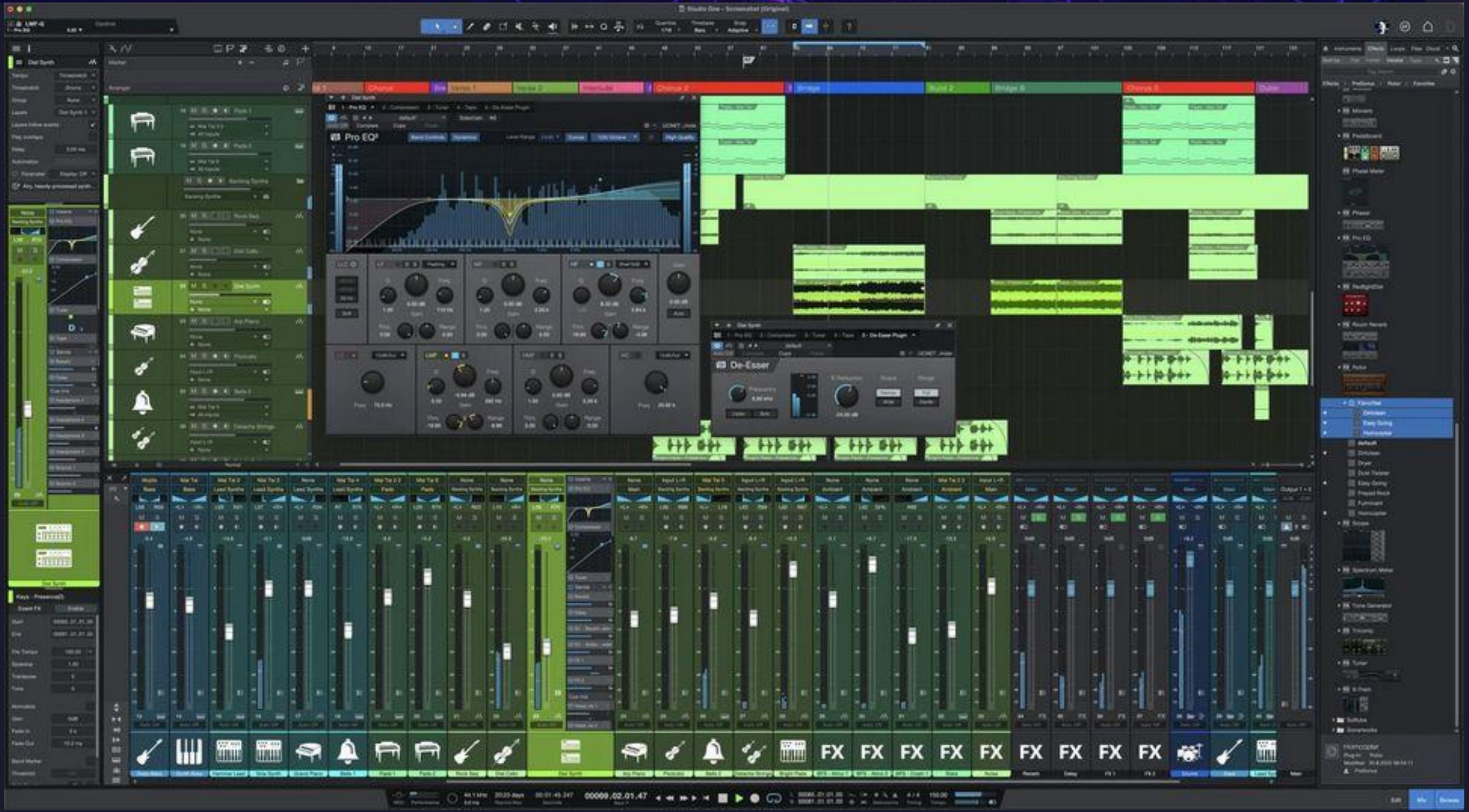
Asheville, NC

About me

- 42-year career as an enterprise Information Architect (IA) and enterprise content systems and knowledge management strategist and content platform development lead
- Built multiple enterprise CMSs and CPDs since the early 90s. Managed 60 million pages of content at IBM in more than for dozen national languages for 3500 products, and thousands of content creators – more than 7M visits per week.
- Cross-trained at IBM as a senior software engineer; designed and built multiple generations of intelligent content supply chains with multiple patents and disclosures
- AI/ML experience dating back to the early 90s, Expert Systems, Small Talk, LISP, and IBM Watson
- Mentored by the inventor of structured markup language. Formed the XML team at IBM that invented DITA XML in the late 90s

ChatGPT is incredibly useful for a myriad of uses, including conceptual learning

- Prosounus Studio One – digital audio workstation (the “light” edition >500-page user guide)
- DAWs are the Photoshop of music production
- Difficult to learn - involves hardware and software
- Used ChatGPT to learn concepts then the user guide to learn the details
- However, ChatGPT doesn’t know about Version 6 or can tell it apart from earlier versions



Search-augmented chatbot demo

Large Language Models (LLMs)

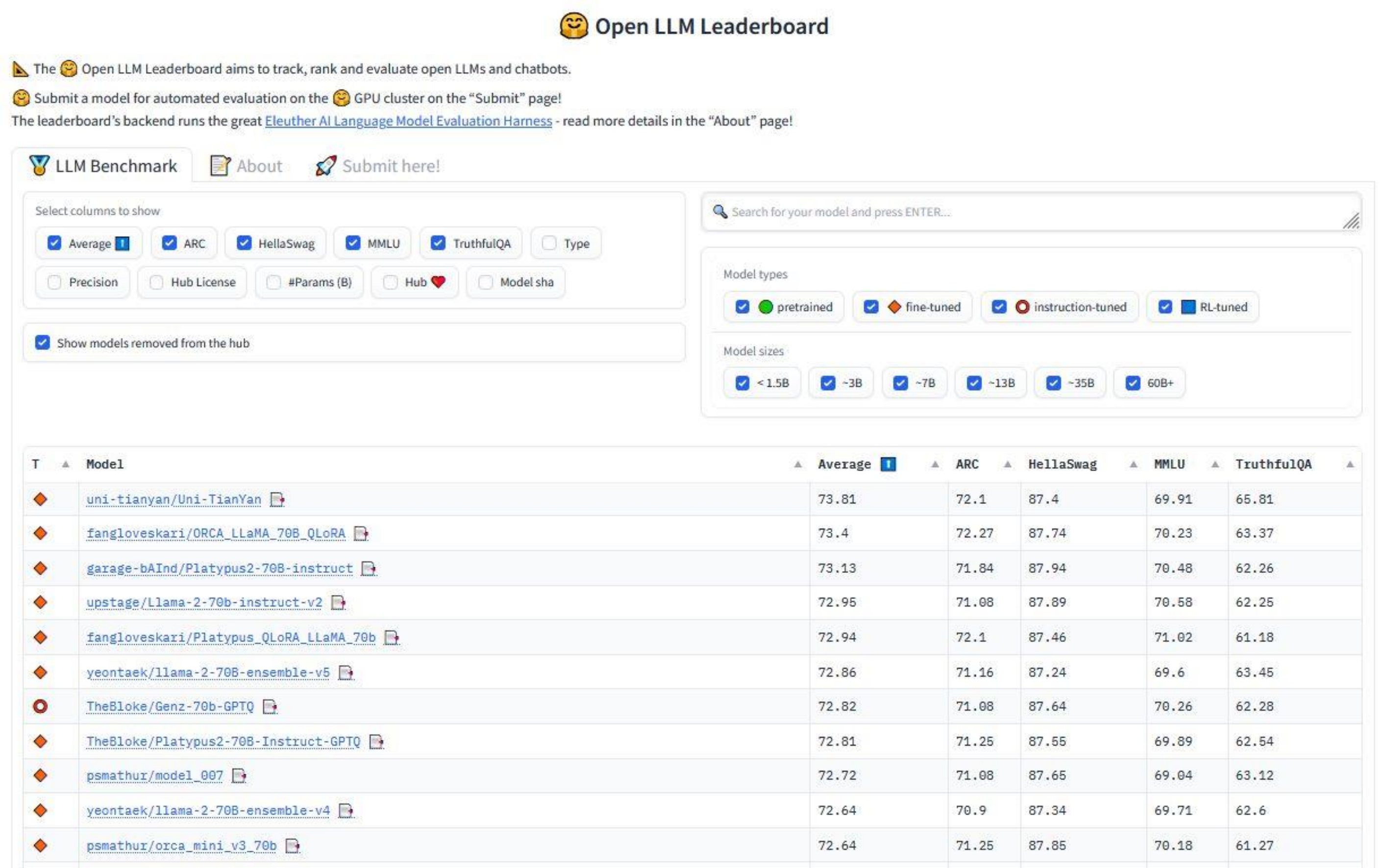
ChatGPT is only one of dozens of large language models (LLMs). LLMs have their roots in language localization

- BERT (Bidirectional Encoder Representations from Transformers) - developed by Google, BERT is trained on a large corpus of text and is fine-tuned for various natural language processing tasks such as sentiment analysis, question-answering, and named entity recognition.
- XLNet - Developed by Carnegie Mellon University and Google AI, XLNet is a permutation-based language model that outperforms BERT and GPT on several benchmark datasets.
- GPT-3 (Generative Pretrained Transformer 3) - developed by OpenAI, GPT-3 is a transformer-based language model that is capable of performing a wide range of natural language processing tasks and is known for its human-like text generation capabilities. Owned by OpenAPI, founded by Peter Thiel, Reid Hoffman, and Elon Musk.
- ELMo (Embeddings from Language Models) - developed by Allen NLP, ELMo is a deep contextualized word representation that models both complex characteristics of word use (e.g., syntax and semantics) and how these uses vary across linguistic contexts (e.g., to model polysemy).

…and most recently, BARD from Google. Bard is a Google framework for creating and deploying machine learning models using TensorFlow, an open-source library for ML development. It is much more powerful than its predecessor, BERT, and requires knowledge of Python and full-stack development skills.

There are dozens of open LLMs

- Not all LLMs are created equal
  - Some don’t permit the ingestion and training using your domain-specific semantic knowledge assets
  - Some don’t support the addition of new concept embeddings
  - Some permit fine-tuning, others don’t
  - Some accept prompt prefixed for supervised and unsupervised training.



🤗 Open LLM Leaderboard

📐 The 🤗 Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

🤗 Submit a model for automated evaluation on the 🤗 GPU cluster on the “Submit” page!
The leaderboard’s backend runs the great Eleuther AI Language Model Evaluation Harness - read more details in the “About” page!

Select columns to show

☑ Average ⬆️ ☑ ARC ☑ HellaSwag ☑ MMLU ☑ TruthfulQA ☐ Type

☐ Precision ☐ Hub License ☐ #Params (B) ☐ Hub ❤️ ☐ Model sha

☑ Show models removed from the hub

🔍 Search for your model and press ENTER...

Model types

☑ 🟢 pretrained ☑ 🔶 fine-tuned ☑ ⭕ instruction-tuned ☑ 🟦 RL-tuned

Model sizes

☑ < 1.5B ☑ ~3B ☑ ~7B ☑ ~13B ☑ ~35B ☑ 60B+

T	Model	Average ⬆️	ARC	HellaSwag	MMLU	TruthfulQA
🔶	uni-tianyan/Uni-TianYan	73.81	72.1	87.4	69.91	65.81
🔶	fangloveskari/ORCA_LLaMA_70B_QLoRA	73.4	72.27	87.74	70.23	63.37
🔶	garage-bAInd/Platypus2-70B-instruct	73.13	71.84	87.94	70.48	62.26
🔶	upstage/Llama-2-70b-instruct-v2	72.95	71.08	87.89	70.58	62.25
🔶	fangloveskari/Platypus_QLoRA_LLaMA_70b	72.94	72.1	87.46	71.02	61.18
🔶	yeontaek/llama-2-70B-ensemble-v5	72.86	71.16	87.24	69.6	63.45
⭕	TheBloke/Genz-70b-GPTQ	72.82	71.08	87.64	70.26	62.28
🔶	TheBloke/Platypus2-70B-Instruct-GPTQ	72.81	71.25	87.55	69.89	62.54
🔶	psmathur/model_007	72.72	71.08	87.65	69.04	63.12
🔶	yeontaek/llama-2-70B-ensemble-v4	72.64	70.9	87.34	69.71	62.6
🔶	psmathur/orca_mini_v3_70b	72.64	71.25	87.85	70.18	61.27

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

The problem with generative LLMs on their own

Current LLMs are not reliable, explainable, or responsible. *Business Insider* recently said LLMs are getting SICKER by the day. LLMs, on their own, lack:

- Stability
- Inferencing
- Content currency and consistency
- Knowledge engineering and content modeling
- Entity relationships
- Reasoning and referential integrity

That's because LLMs, by themselves, have no ability to infer, have no persistent referential integrity, lack any capability to reason, nor have any memory, In short, stochastic AI models cannot explain themselves - we need to make our own models explainable AI (XAI), and the only way to do that is by augmenting these stochastic LLMs with symbolic AI using engineered ontologies and knowledge graphs that capture persistent relationships that can be queried and improve over time.

LLM risks and limitations

Understanding the limitations of LLMs will help you apply them effectively

- Public LLMs, such as ChatGPT do not have current content. ChatGPT, for example, generally uses content only through 2021. For most of us, a lack of content currency and a significant limitation
- LLMs cannot create original content with intent and cannot verify facts. LLMs are it's nowhere near ready to replace content creation jobs anytime soon
- LLMs do not know logical content boundaries and dependencies – They only tokenize words and chunks
- They are not experts in your domain, your business and technical language, your users, or your content models

…however, we have the technology and methods needed to address these deficiencies

Bias Liability

Misinformation

Intellectual Property

Security Propaganda

Copyright

Privacy Currency

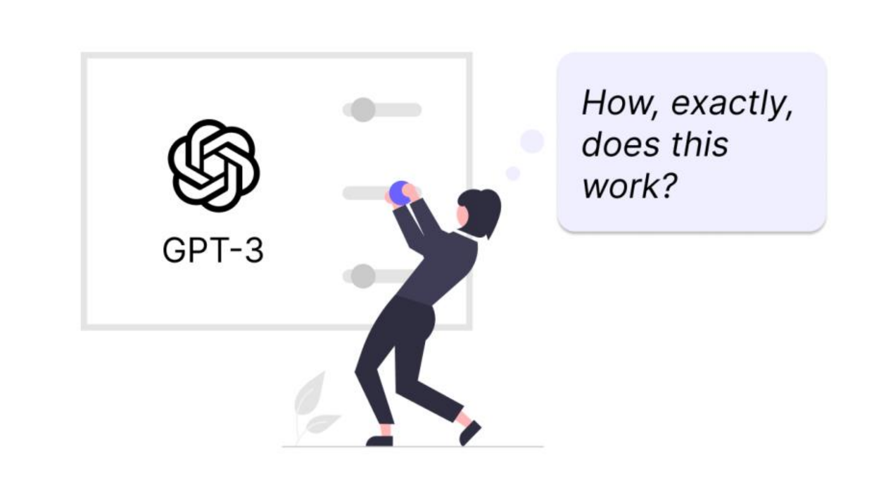
Phishing Cyberthreat

Utterly confident and convincing (right or wrong)

How LLMs work

LLMs use stochastic techniques and services

- Public models ingest billions of words
- Typically use a neural network called a *transformer*
- Learns language patterns, weighting algorithms
- Uses a *predictive* model – generates text that is similar to the text it was trained by predicting the next word in a sentence given the relative distance of the words that come before it
- When predicting, the model considers all possible words it could use and calculates the probability of each. It then selects the word with the highest probability and adds it to the output. This continues until the model completes the sentence, paragraph, or even the entire article



How LLMs really work under the covers

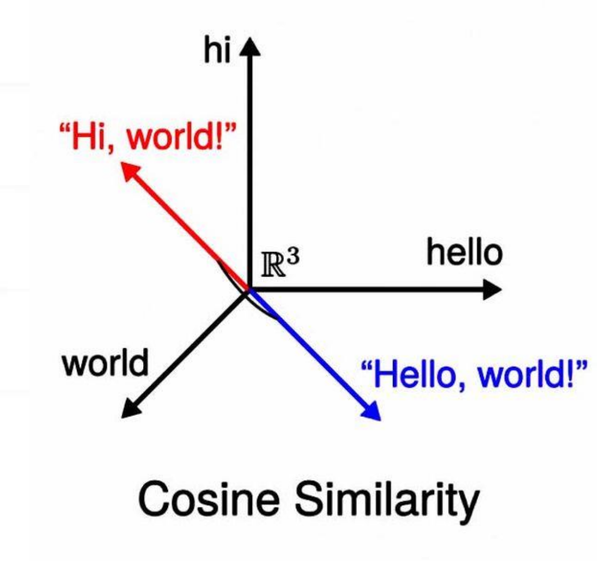
Cosine similarity

ChatGPT uses cosine similarity as a measure of similarity between two vectors in certain tasks, such as natural language processing (NLP) tasks like text classification, clustering, and information retrieval.

What matters with vectors is distance + trajectory. Cosine similarity determines the cosine of the angle between two non-zero vectors in a multi-dimensional space. In NLP, we often represent text as vectors of word frequencies or word embeddings, and cosine similarity can be used to measure the *similarity* between *two pieces of text* based on their vector representations.

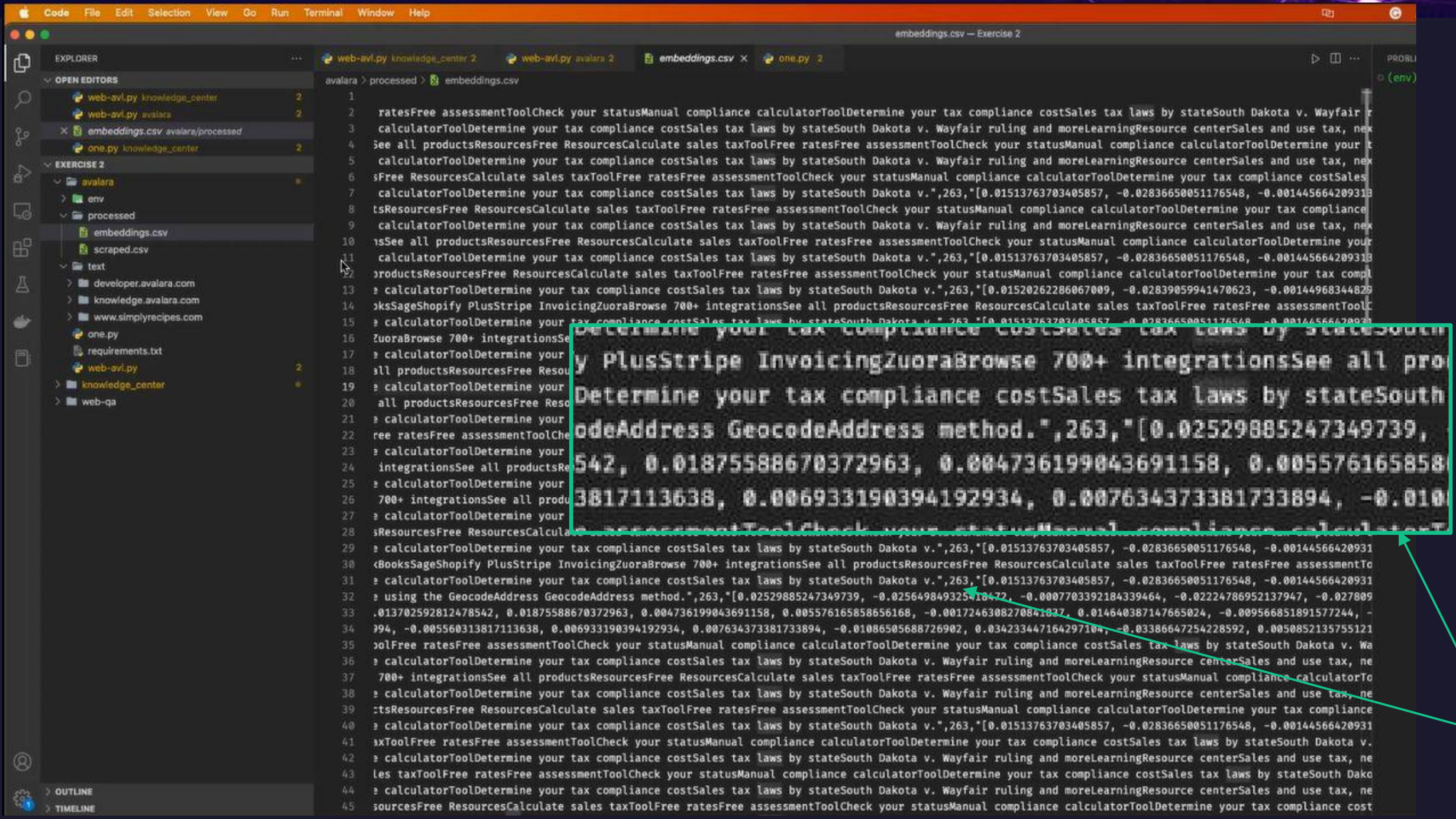
For example, if we have two vectors representing two pieces of text, we can calculate their cosine similarity as the dot product of the two vectors divided by the product of their magnitudes. The resulting value ranges from -1 to 1, with 1 indicating that the two vectors are identical, 0 indicating that they are orthogonal or unrelated, and -1 indicating that they are diametrically opposed.

Linear algebra



Note: ChatGPT's temperature parameter explains why you get different answers to the same question. It determines how often lower-ranked words will be used for variety

What OpenAI does with our ingested content



```
ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair r
calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
See all productsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your t
calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
sFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance costSales
calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
tsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance
calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
nsSee all productsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine you
calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
roductsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compl
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01520262286067009, -0.02839059941470623, -0.00144968344829
oksSageShopify PlusStripe InvoicingZuoraBrowse 700+ integrationsSee all productsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolC
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
ZuoraBrowse 700+ integrationsSe
e calculatorToolDetermine your
all productsResourcesFree Resou
e calculatorToolDetermine your
 all productsResourcesFree Reso
e calculatorToolDetermine your
ree ratesFree assessmentToolChe
e calculatorToolDetermine your
 integrationsSee all productsRe
e calculatorToolDetermine your
 700+ integrationsSee all produ
e calculatorToolDetermine your
sResourcesFree ResourcesCalcula
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
<BooksSageShopify PlusStripe InvoicingZuoraBrowse 700+ integrationsSee all productsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentTo
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
e using the GeocodeAddress GeocodeAddress method.",263,"[0.02529885247349739, -0.02564984932518472, -0.0007703392184339464, -0.02224786952137947, -0.027809
.013702592812478542, 0.01875588670372963, 0.004736199043691158, 0.005576165858656168, -0.0017246308270841827, 0.014640387147665024, -0.009566851891577244, -
994, -0.005560313817113638, 0.006933190394192934, 0.007634373381733894, -0.01086505688726902, 0.03423344716429710, -0.03386647254228592, 0.0050852135755121
olFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wa
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
 700+ integrationsSee all productsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorTo
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
tsResourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.",263,"[0.01513763703405857, -0.02836650051176548, -0.00144566420931
axToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v.
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
les taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dako
e calculatorToolDetermine your tax compliance costSales tax laws by stateSouth Dakota v. Wayfair ruling and moreLearningResource centerSales and use tax, ne
sourcesFree ResourcesCalculate sales taxToolFree ratesFree assessmentToolCheck your statusManual compliance calculatorToolDetermine your tax compliance cost
```

```
y PlusStripe InvoicingZuoraBrowse 700+ integrationsSee all pro
Determine your tax compliance costSales tax laws by stateSouth
odeAddress GeocodeAddress method.",263,"[0.02529885247349739,
542, 0.01875588670372963, 0.004736199043691158, 0.00557616585
3817113638, 0.006933190394192934, 0.007634373381733894, -0.010
```

Uses NLP techniques on ingestion

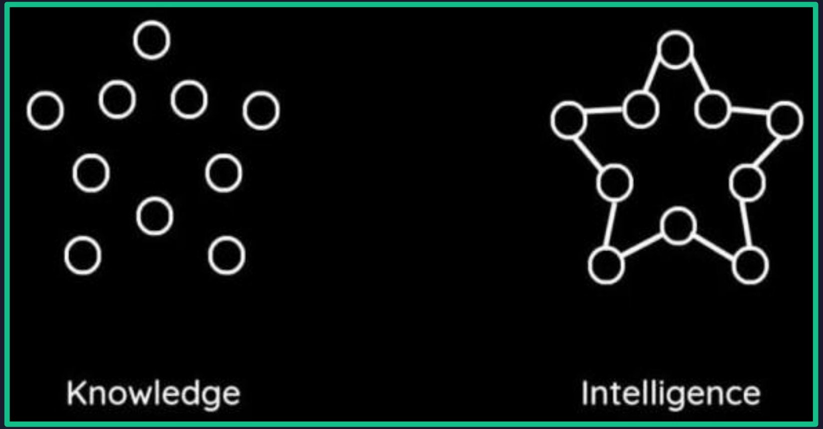
- Tokenization
- Stemming and Lemmatization
- Stop Words Removal
- Keyword Extraction
- Outline parts of speech
- Determines relevancy/Importance
- Word Embeddings
- Sentiment Analysis
- Topic Modelling
- Text Summarization
- Named Entity Recognition

Then converts to vectors and combined with vectors of the corpus of DB and the nearest distance is determined.

LLMs such as ChatGPT aren’t knowledge generators – they are ***auto-regressive* models**

LLMs work on predictive relationships of words. They cannot do *inferencing*, nor understand *concepts*

- Limited by data it was trained on, using an AI model that operates based on statistical patterns in text, rather than a true understanding of the world.
- They cannot disambiguate well – it relies almost entirely on context.
- They cannot reason about complex or abstract concepts in the same way that a human can, and they *can make mistakes* or *generate inappropriate responses (aka, hallucinations)*, especially when dealing with more nuanced or *context-specific* information.
- The quality and accuracy of its inferencing depend on the quality and breadth of the training data it was exposed to, as well as the complexity of the inferencing task. It often makes mistakes and arrives at incorrect conclusions, especially when dealing with complex or novel information.



LLMs are more akin to an auto-completion tools that uses patterns in language to generate coherent and plausible responses, rather than as a model with a true understanding of the world and concepts. That’s what well-architected Knowledge Graphs provide.

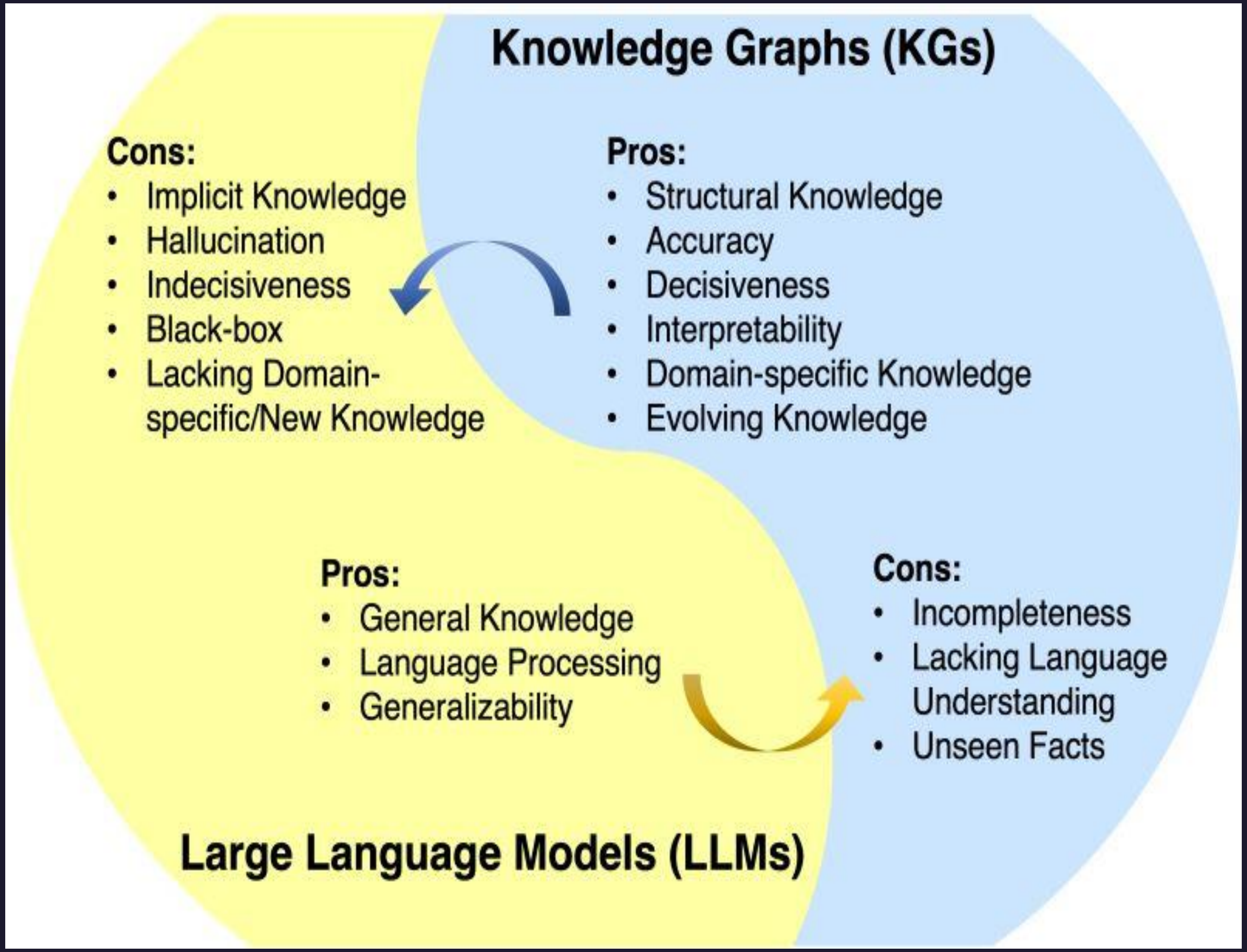
Beyond pedestrian use of ChatGPT and LLMs

- We need to get beyond simply using LLMs as they are provided OOTB and provide our users with reliable content, including:
  - Current content
  - New and updated content
  - Conforming and consistent content
  - Trustworthy content - verifiable
  - Static *and* dynamic content (CaaS)
  - Human and event-driven assistance
  - Personified and personalized content
  - Progressive disclosure
  - Precision answers
  - Reusable content
  - Complex scenario assistance

Requires

Managed content and knowledge models (but how?)

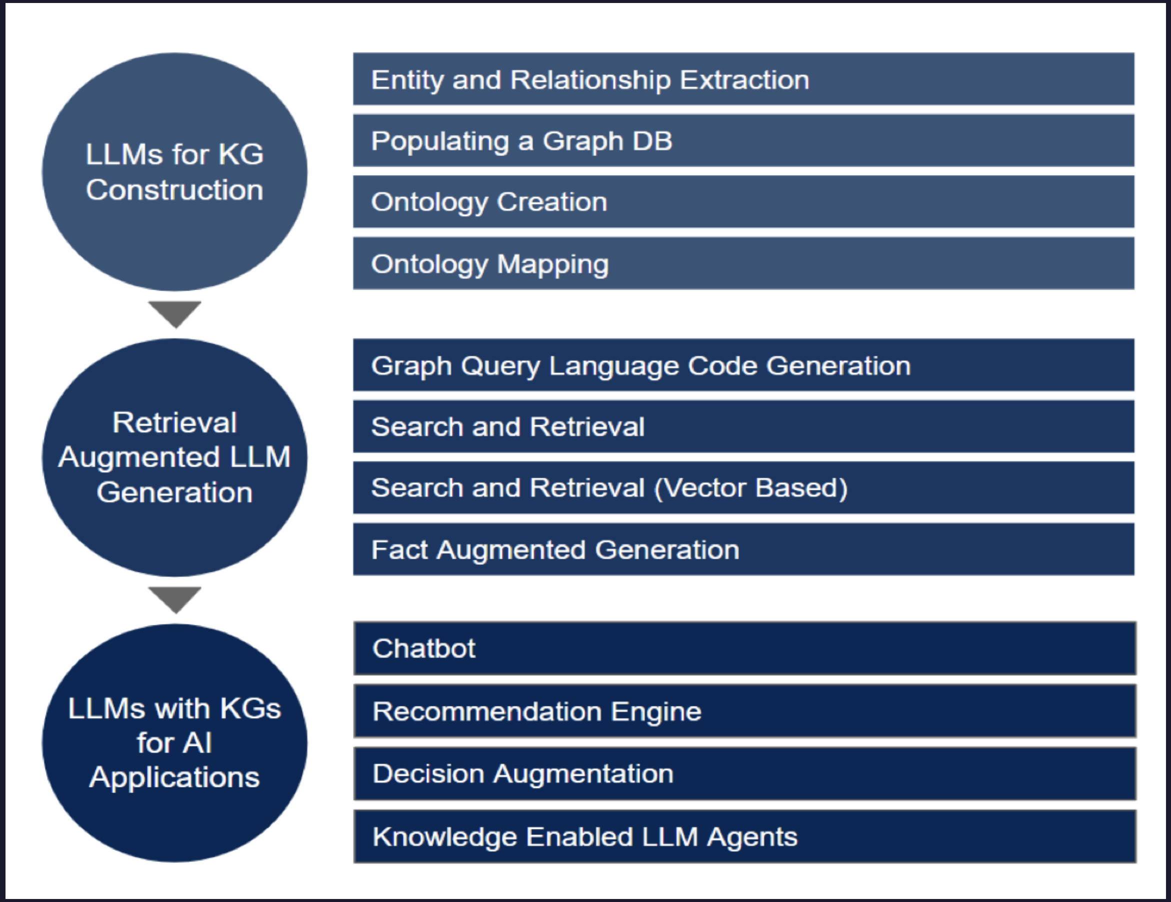
The Case for Augmenting LLMs with a Knowledge Graph



Source: **Unifying Large Language Models and Knowledge Graphs: A Roadmap** https://www.arxiv-vanity.com/papers/2306.08302/

- Most LLMs employ a stochastic (predictive) model
  - We're aiming to reduce friction with our product offerings, improve time-to-value, and provide reliable self-service.
  - Guessing with either no resulting answers or wrong answers of any significant percentage is unacceptable.
  - Most of us work with large volumes of domain-specific content. While we need a pre-trained LLM, we often don't need the content that was used to train it; we need only our own content.
- Deterministic models, such as knowledge graphs, are far more predictable – combine both.
  - Especially when based on intentionally curated taxonomies, ontologies, and knowledge graphs that many companies have developed or are in the process of developing.
  - Deterministic models are explainable (XAI) and can provide the referential integrity that users demand.
  - Achieving highly effective generative AI requires combining stochastic LLMs with deterministic knowledge graphs

Combine LLMs with knowledge graphs



LLMs for KG Construction

- Entity and Relationship Extraction
- Populating a Graph DB
- Ontology Creation
- Ontology Mapping

Retrieval Augmented LLM Generation

- Graph Query Language Code Generation
- Search and Retrieval
- Search and Retrieval (Vector Based)
- Fact Augmented Generation

LLMs with KGs for AI Applications

- Chatbot
- Recommendation Engine
- Decision Augmentation
- Knowledge Enabled LLM Agents

Large language models and knowledge graphs compared

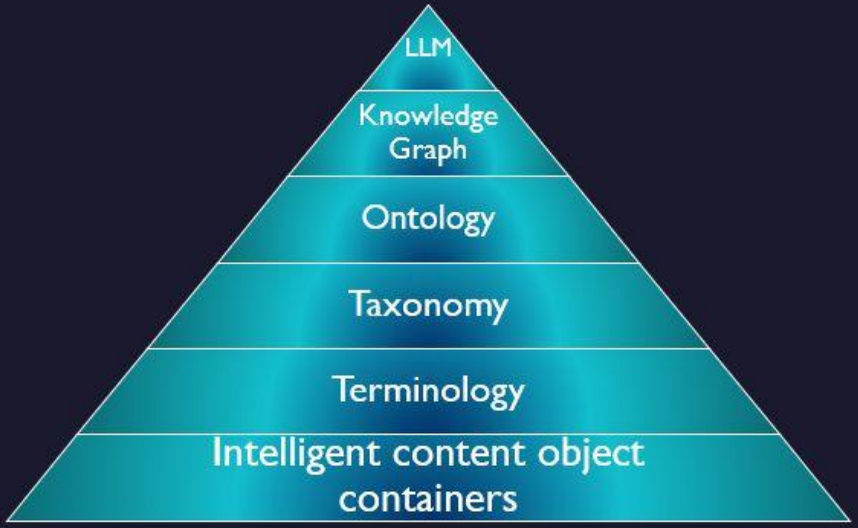
- Large Language Models (LLMs) are based on correlations
- LLMs are trained using unsupervised learning
- LLMs are black-box models
- LLMs fall short of capturing and accessing referencable factual knowledge

- KGs provide semantic incl. causal relationships
- Knowledge models are built using supervised learning
- Knowledge Graphs (KGs) are based on explicit, structured knowledge models
- KGs provide referencable, rich factual knowledge

It’s not about programming or data, it’s about managing *knowledge* and *content*

The path to reliable, accurate, trustworthy, responsible, and explaining AI (XAI) is through LLM augmentation and *intelligent content*

- Develop and augment LLMs with domain-specific symbolic AI assets the form of taxonomies, ontologies, and knowledge graphs
- Componentize content and make it consistent using computational linguistic technology
- Make content “intelligent” by making it self-describing
  - Typed containers and objects
  - Labeled with controlled vocabularies
  - Algorithmically predictable and reliable



LLM

Knowledge Graph

Ontology

Taxonomy

Terminology

Intelligent content object containers

What is explainable AI (XAI)?

Generative AI cannot explain its results

- GAI generates human-like responses to user inputs. However, it is *not* an explainable AI model (XAI). GAI rarely produces the same answer twice. Just ask ChatGPT – it admits that it's not XAI

- Explainable AI refers to models that are designed to provide clear and understandable explanations of their decision-making processes

- While GAI can provide insight into how it generates its responses by examining the input data and the neural network weights, it is not designed to provide clear and understandable explanations of its decision-making process. Therefore, it is not considered an explainable AI model *by itself*.

XAI is especially important in cases where decisions have significant real-world consequences, such as in medical diagnosis or financial forecasting.

The hard truth…

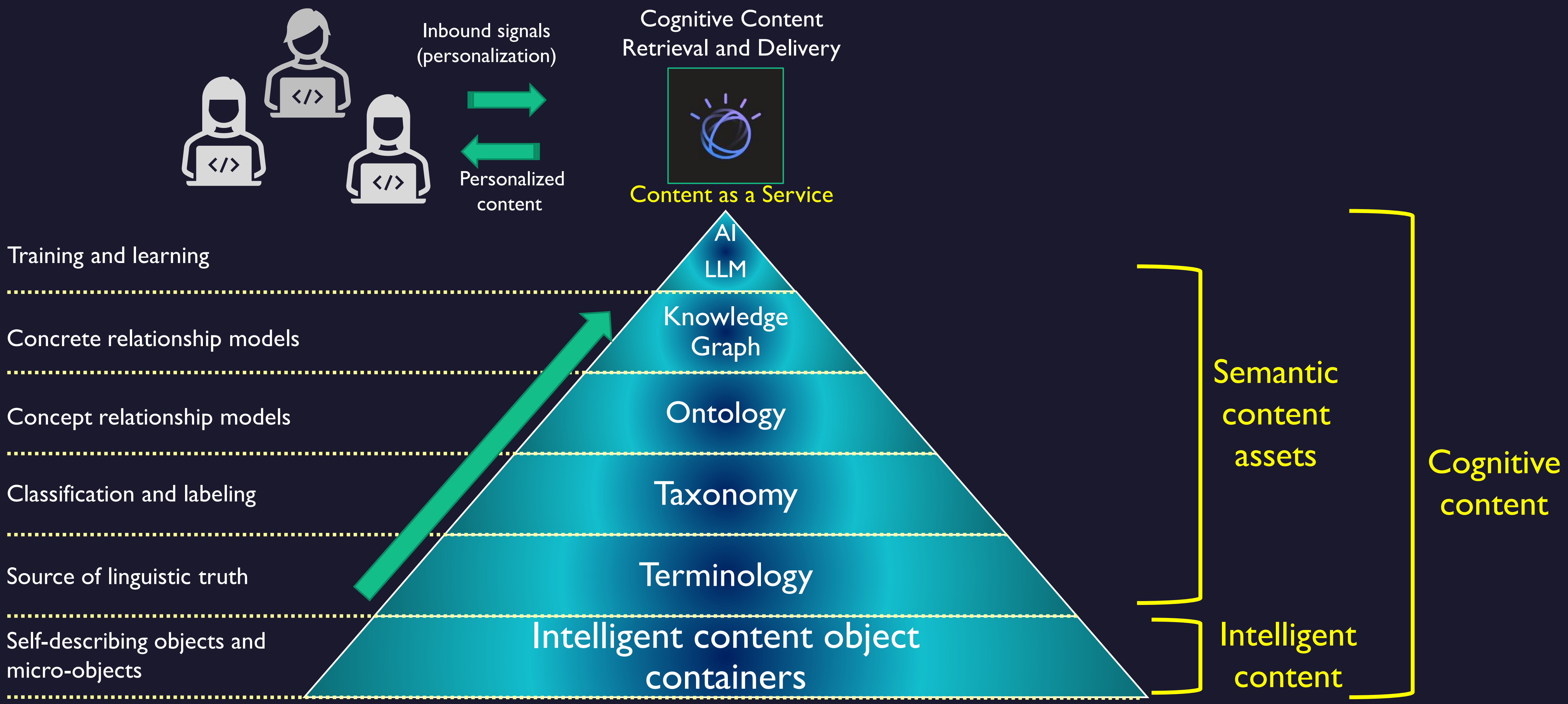
Public LLM models are not the solution for user assistance

- Public LLM content such as ChatGPT is already stale by two years
- They cannot be made current; no amount of use, crowdsourcing, or manual training use will accomplish that
- They don't have access to billions of updates and additions, and LLMs don't share

For critical information, the only viable answer is to train an LLM using your own content corpus exclusively

- Choose an LLM, ingest your own content, and keep it fresh and avoid using public content
- Train your LLM, at scale, using automated means (auto-classification)
- Provide referential integrity by augmenting prompts by querying your own domain-curated knowledge graph and provide source links with the results

The semantic content maturity model (SCMM)



Inbound signals (personalization)

Personalized content

Cognitive Content Retrieval and Delivery

Content as a Service

AI

LLM

Knowledge Graph

Ontology

Taxonomy

Terminology

Intelligent content object containers

Training and learning

Concrete relationship models

Concept relationship models

Classification and labeling

Source of linguistic truth

Self-describing objects and micro-objects

Semantic content assets

Intelligent content

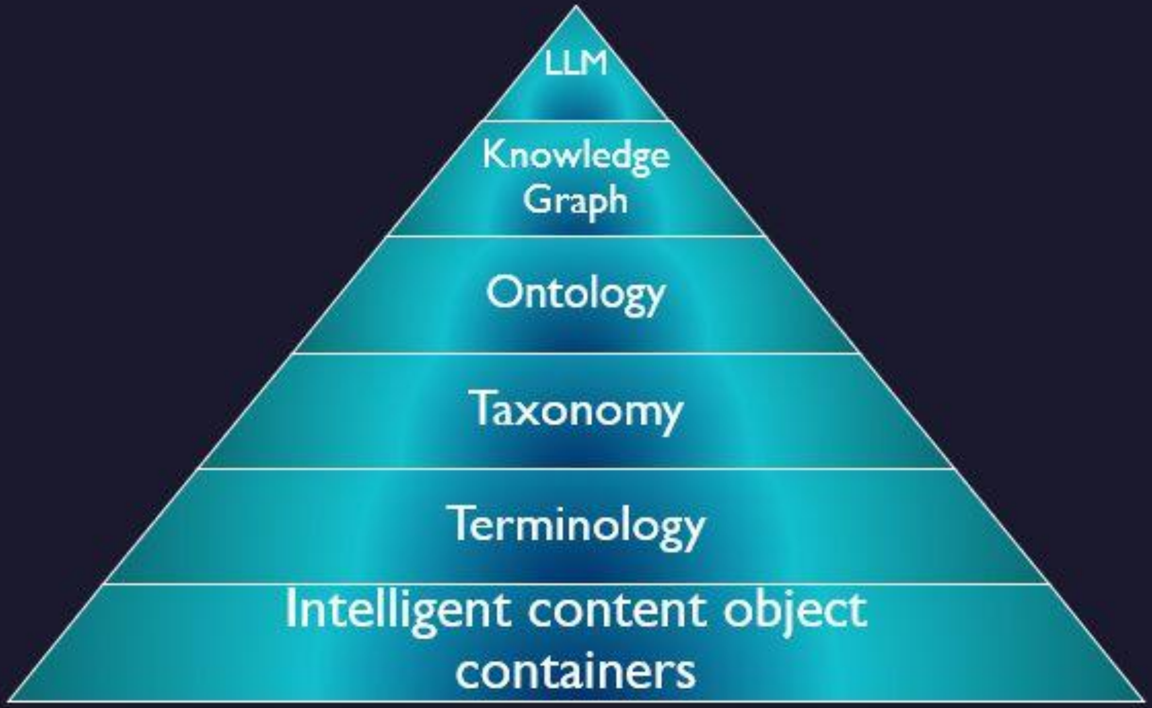
Cognitive content

Visit https://thinkingdocumentation.com/blog For a detailed overview of the semantic content maturity model

Train your LLM's brain with intelligent content

Intelligent content is modular, structured, reusable, separates format from presentation, and is semantically enriched such that the content is highly predictable for machine processing and automation.

- You know the old saying – "GIGO"
- School these LLMs - teach them your domain vocabularies and semantic models
- Apply the knowledge that's missing for accuracy, reliability, and trustworthiness
- Did you know that you can train many LLMs using your own domain-specific ontologies?



LLM

Knowledge Graph

Ontology

Taxonomy

Terminology

Intelligent content object containers

Semantic content maturity model

(M. Iantosca, 2022)

Taxonomies vs. Ontologies vs. Knowledge Graphs

Taxonomies

- Static
- Classification of things - not strings!
- Lists or hierarchical relationships
- The problem with taxonomies alone: They don't describe relationships outside of the list or across different branches of a tree

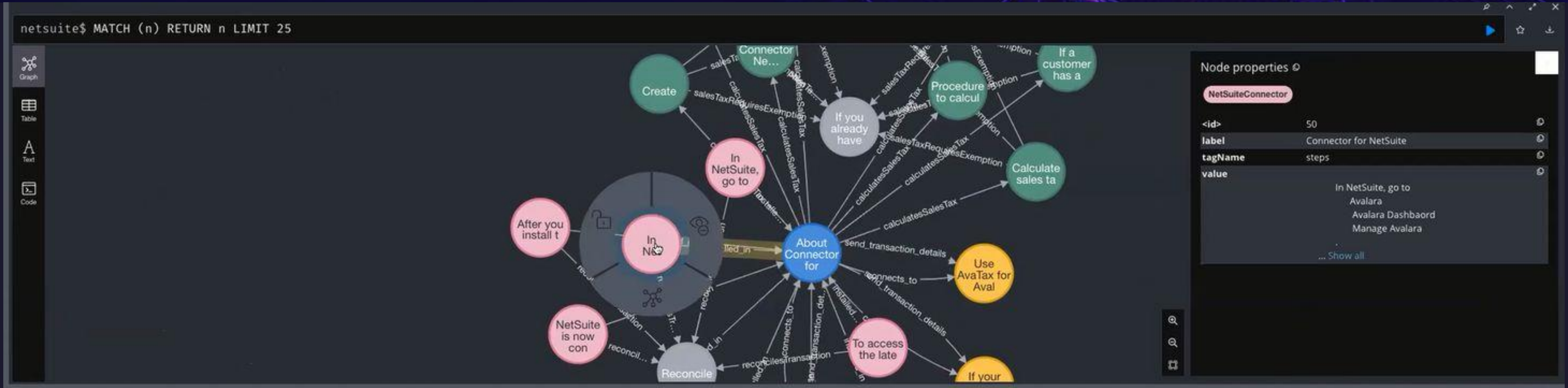
Ontologies

- Describes subclasses and super classes and their relationships for specific domains (e.g. tax compliance)
- Ontologies only model *general* types of things that share certain properties but don't reference real-world objects
- Defines concepts and the properties that describe them
- Serves as the schema for a knowledge graph. You can use an existing ontologies or develop a custom one for a specific domain.

Knowledge Graphs

- Adds a data layer for a specific instance
- Extends an existing ontology; represents real-world objects and their relationships
- Grows and improves over time with additional intelligence
- Links all your data together, at scale - structured or unstructured
- Adds extended data about individual entities (called *edge data*)
- Used by AI/ML to recognize patterns (cognitive intelligence) and predict new relationships (inferencing)

We can then graph them for advanced semantic query - pre-search before invoking the LLM and post-generation for verification

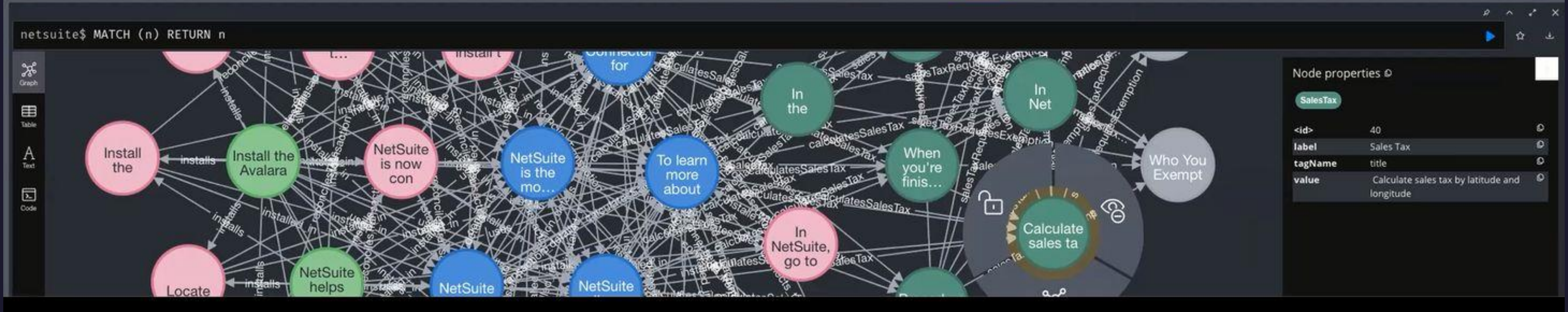


```
netsuite$ MATCH (n) RETURN n LIMIT 25
```

Node properties

NetSuiteConnector

<id>	50
label	Connector for NetSuite
tagName	steps
value	In NetSuite, go to Avalara Avalara Dashbaord Manage Avalara … Show all



```
netsuite$ MATCH (n) RETURN n
```

Node properties

SalesTax

<id>	40
label	Sales Tax
tagName	title
value	Calculate sales tax by latitude and longitude

A sample model for augmenting a generative chatbot

Augmenting large language models with knowledge graphs for effective, responsible, and explainable AI (XAI) – M. Iantosca

Full text:

https://thinkingdocumentation.com/downloads

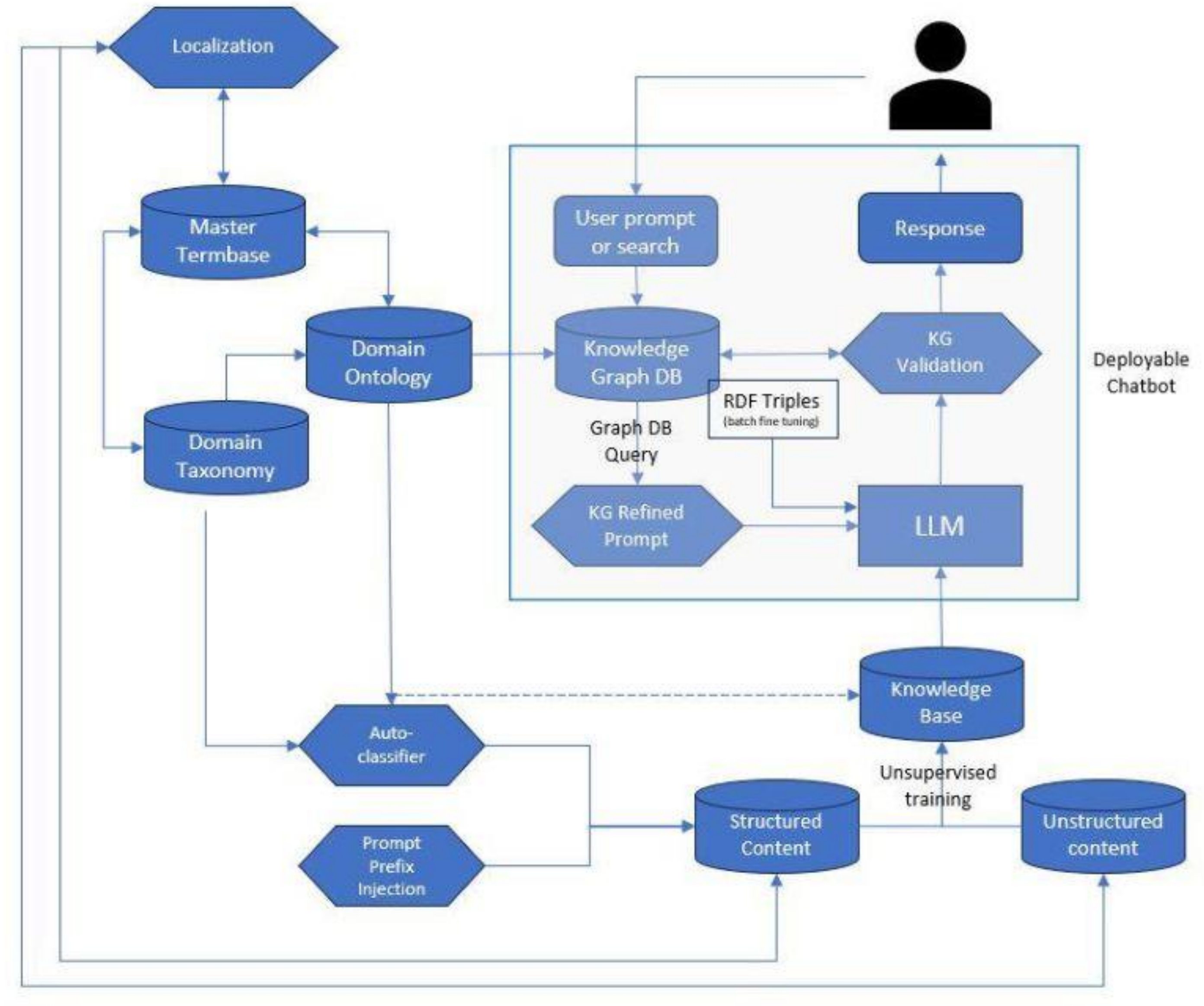


Figure 1: KG-driven generative pattern

Tony Seal's (UBS) implementation methods:

https://www.linkedin.com/posts/tonyseale_knowledgegraph-llms-datascience-activity-7095675217133359104-EsxC?utm_source=share&utm_medium=member_desktop

Generative AI is not cognitive content, nor a complete cognitive content supply chain

Cognitive content and a cognitive content supply chain is a strategy, an architecture, and an operational model that enables dynamic, machine-based retrieval, assembly, and delivery of non-linear content objects to provide humans and machines with knowledge that is based on predictive relationships between content objects and inbound signals.

mji

Advancing from reactive, *failure-mode* content to hyper-personalized, *pro-active* and assistive content

Summary

We’re standing at the precipice of another major content inflection point, moving beyond intelligent content to *cognitive content,* opening up a whole new world of knowledge-driven content applications.

Blog: https://thinkingdocumentation.com/blog

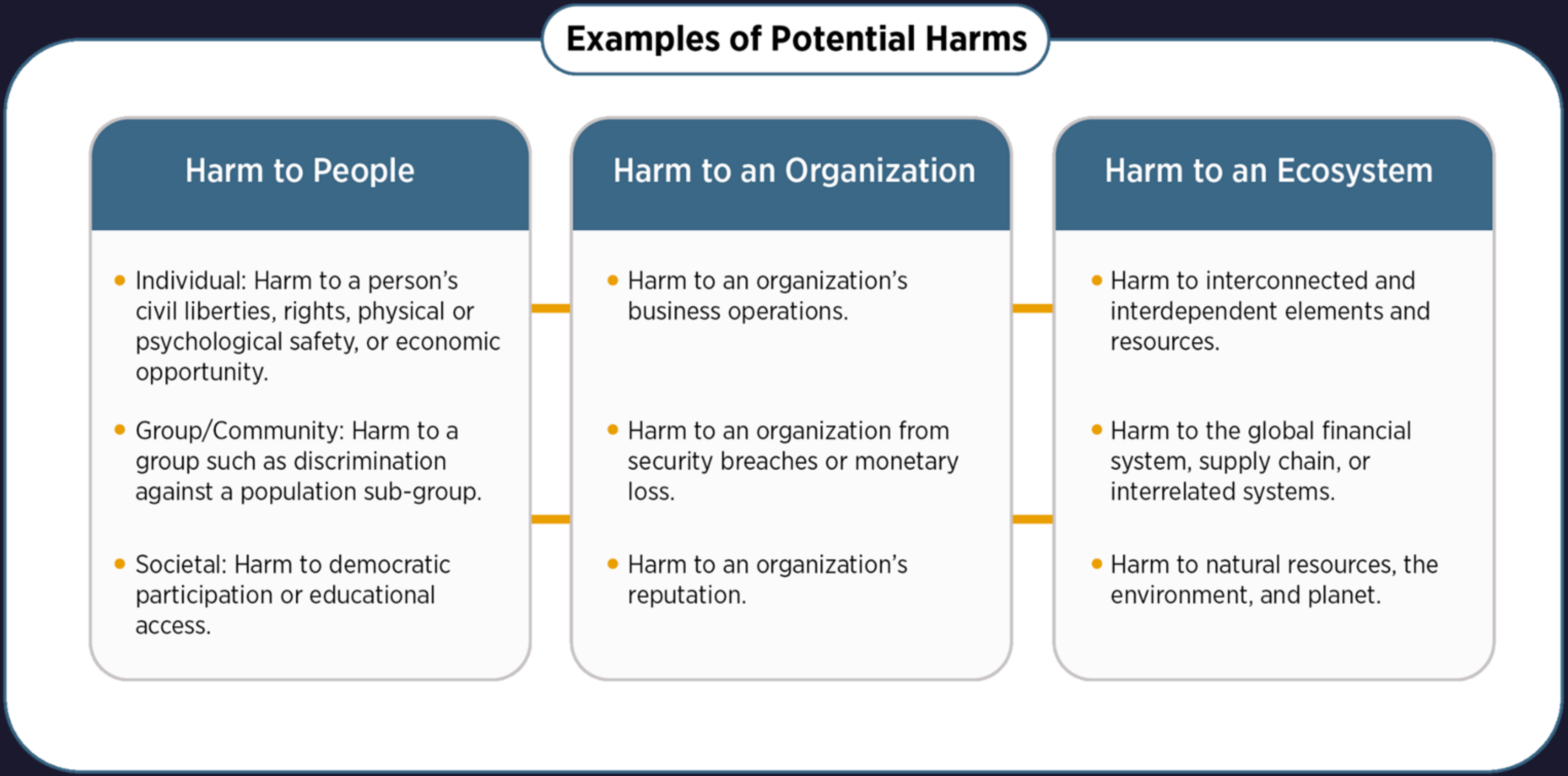
Semantic content graph guild (public discussion forum): https://thinkingdocs.com/ (password= graphs)

Managing Risk

Mitigating risks using NIST’s AI risk management framework (RMF)

"Approaches which enhance AI trustworthiness can reduce negative AI risks”."

NIST
National Institute of Standards and Technology



Examples of Potential Harms

Harm to People

- Individual: Harm to a person’s civil liberties, rights, physical or psychological safety, or economic opportunity.
- Group/Community: Harm to a group such as discrimination against a population sub-group.
- Societal: Harm to democratic participation or educational access.

Harm to an Organization

- Harm to an organization’s business operations.
- Harm to an organization from security breaches or monetary loss.
- Harm to an organization’s reputation.

Harm to an Ecosystem

- Harm to interconnected and interdependent elements and resources.
- Harm to the global financial system, supply chain, or interrelated systems.
- Harm to natural resources, the environment, and planet.

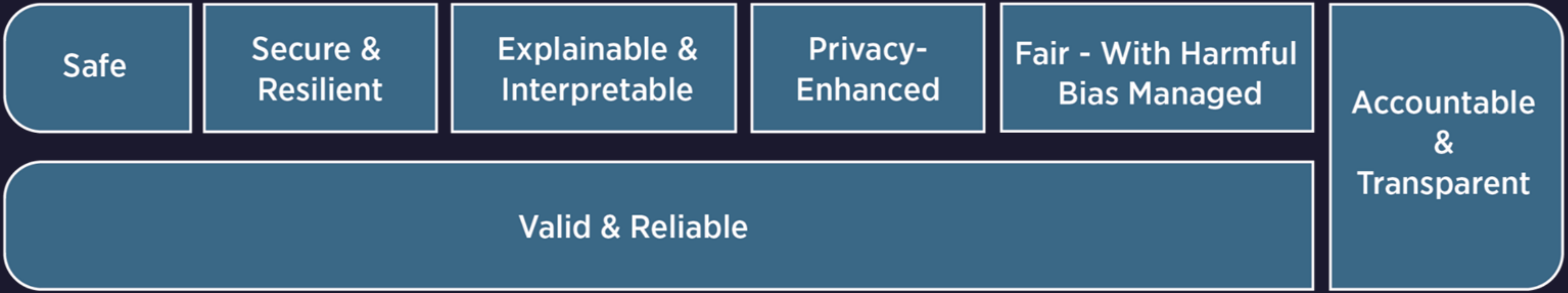
NIST's AI Risk Management Framework

https://airc.nist.gov/AI_RMF_Knowledge_Base/AI_RMF

Characteristics of trustworthy AI

AI Risks and Trustworthiness

https://airc.nist.gov/AI_RMF_Knowledge_Base/AI_RMF/Foundational_Information/3-sec-characteristics



Safe

Secure & Resilient

Explainable & Interpretable

Privacy-Enhanced

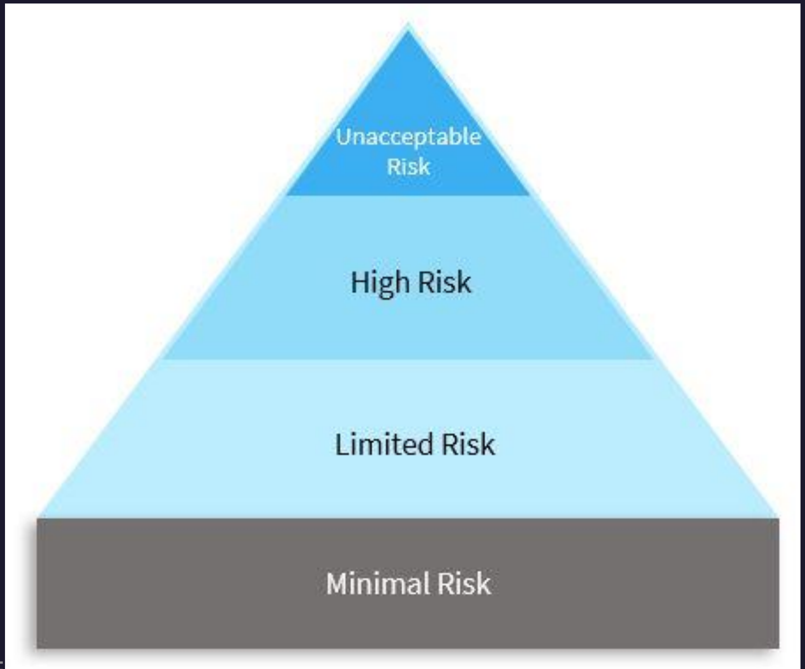
Fair - With Harmful Bias Managed

Accountable & Transparent

Valid & Reliable

The cornerstone of the EU AI Act is a classification system that determines the level of risk an AI technology could pose to the health and safety or fundamental rights of a person.

European Commission

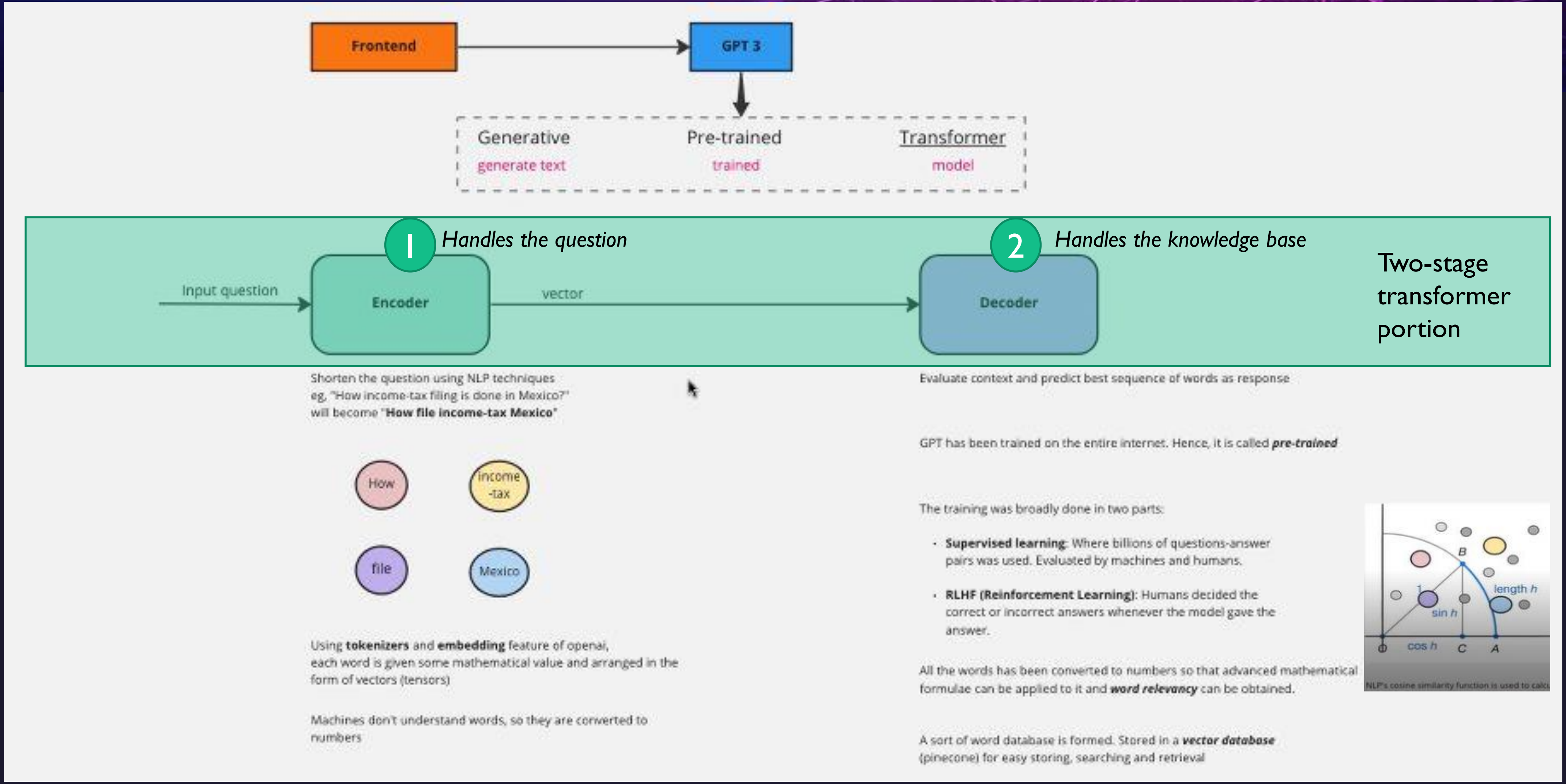


https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

Measuring risk requires a managed program

- Diversity and Robustness of Outputs
- Human Evaluation
- Bias and Fairness **Assessment**
- Adversarial Testing
- Ethical Guidelines and Constraints
- Fine-tuning and Iteration
- Data Analysis
- Legal and Regulatory Compliance
- User Feedback and Reporting
- Third-Party Audits
- Simulations and Scenario Analysis
- Transparency and Explainability
- Red Teaming
- Robustness Testing
- Deployment Monitoring

More about how ChatGPT works



Handles the question

Handles the knowledge base

Two-stage transformer portion

Shorten the question using NLP techniques
eg, "How income-tax filing is done in Mexico?"
will become "**How file income-tax Mexico**"

Using **tokenizers** and **embedding** feature of openai,
each word is given some mathematical value and arranged in the form of vectors (tensors)

Machines don't understand words, so they are converted to numbers

Evaluate context and predict best sequence of words as response

GPT has been trained on the entire internet. Hence, it is called ***pre-trained***

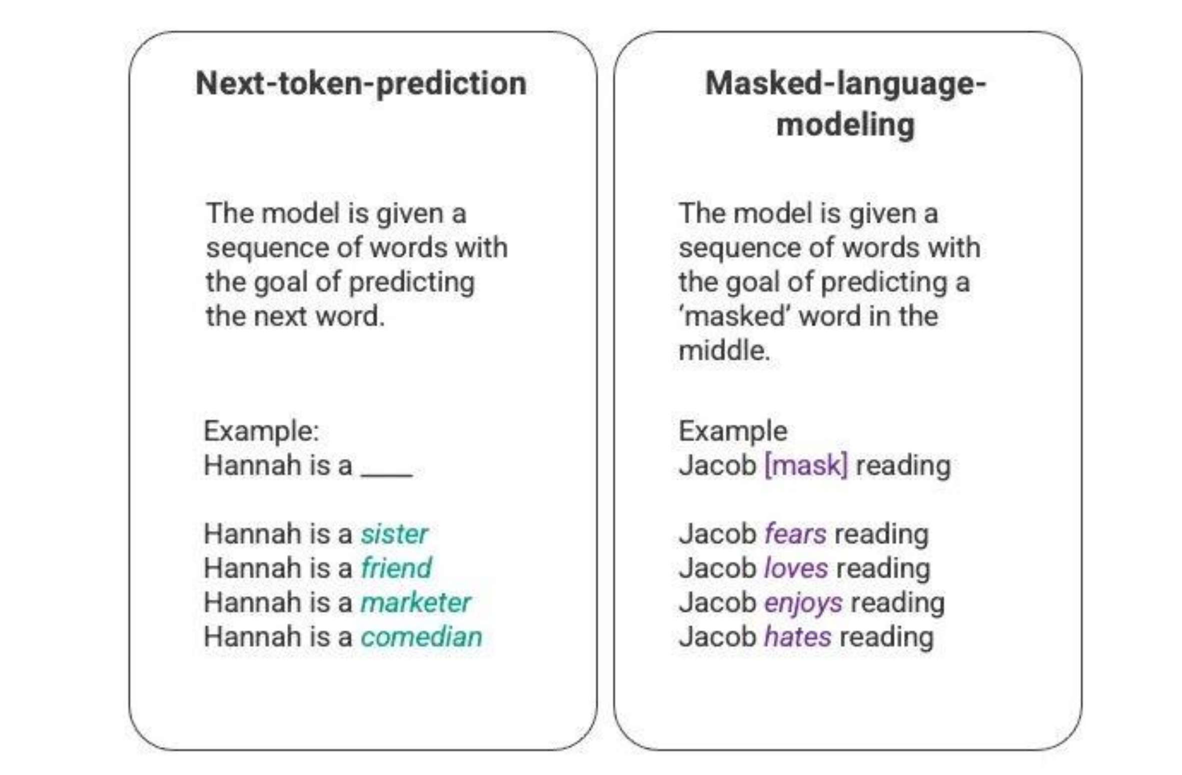
The training was broadly done in two parts:

- **Supervised learning**: Where billions of questions-answer pairs was used. Evaluated by machines and humans.
- **RLHF (Reinforcement Learning)**: Humans decided the correct or incorrect answers whenever the model gave the answer.

All the words has been converted to numbers so that advanced mathematical formulae can be applied to it and ***word relevancy*** can be obtained.

A sort of word database is formed. Stored in a ***vector database*** (pinecone) for easy storing, searching and retrieval

How ChatGPT Works:
A predictive “auto-regressive” model



Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

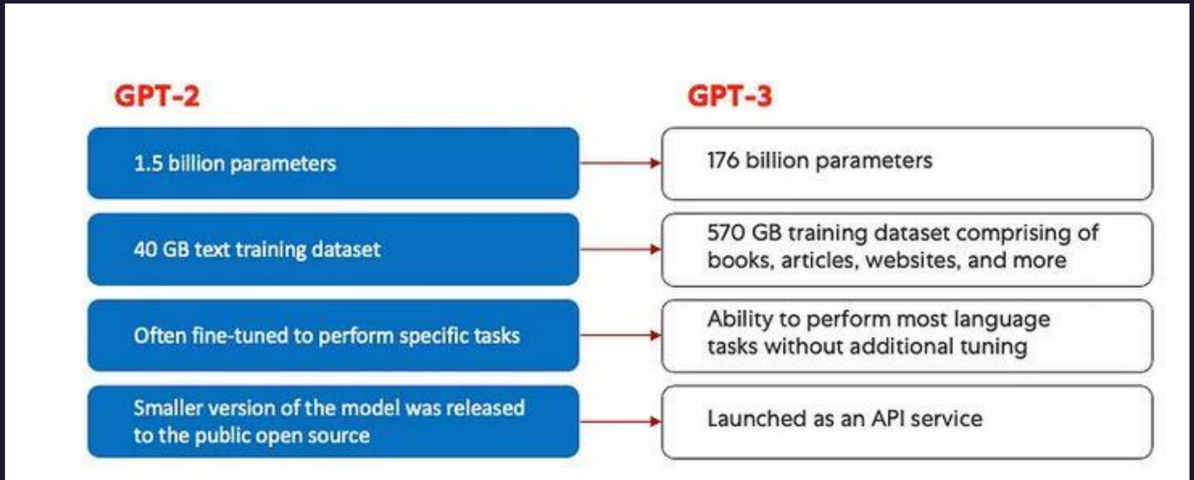
Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

Masked-language-modeling

The model is given a sequence of words with the goal of predicting a ‘masked’ word in the middle.

Example
Jacob [mask] reading

Jacob *fears* reading
Jacob *loves* reading
Jacob *enjoys* reading
Jacob *hates* reading

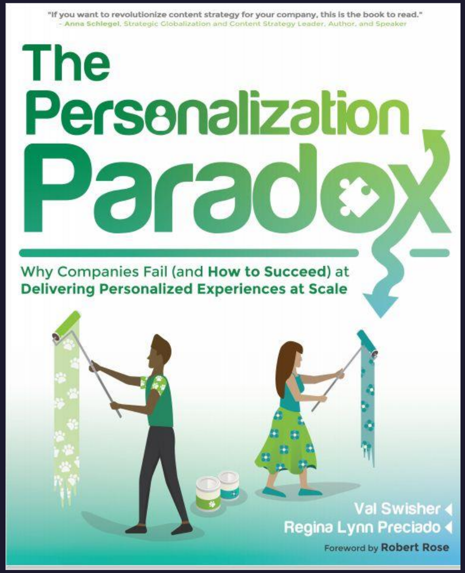


GPT-2	GPT-3
1.5 billion parameters	176 billion parameters
40 GB text training dataset	570 GB training dataset comprising of books, articles, websites, and more
Often fine-tuned to perform specific tasks	Ability to perform most language tasks without additional tuning
Smaller version of the model was released to the public open source	Launched as an API service

In an autoregressive language model, the model generates text one token at a time, based on the previous tokens it has generated. Specifically, the model calculates the probability distribution over the possible next tokens, given the previous tokens, and then samples from that distribution to generate the next token.

A deeper dive into ontology

- An ontology identifies *relationships* between *classes*. It tells the machine what those relationships are for machine retrieval. It is encoded in Ontology Web Language (OWL) which is an XML schema.
- Consists of triples (Subject + Predicate + Object) and stored in a database called a triplestore (also called an RDF store) for the retrieval of triples through semantic queries.
- Enables terabytes of data to be reduced to only a few gigabytes of relevant data allowing more precise and effective semantic search.
- An ontology sets the foundation for a knowledge graph to capture data; it serves as the backbone for a knowledge graph.



"If you want to revolutionize content strategy for your company, this is the book to read."
- Anna Schlegel, Strategic Globalization and Content Strategy Leader, Author, and Speaker

The Personalization Paradox

Why Companies Fail (and How to Succeed) at Delivering Personalized Experiences at Scale

Val Swisher
Regina Lynn Preciado
Foreword by Robert Rose

"Think of an ontology as a taxonomy of taxonomies."

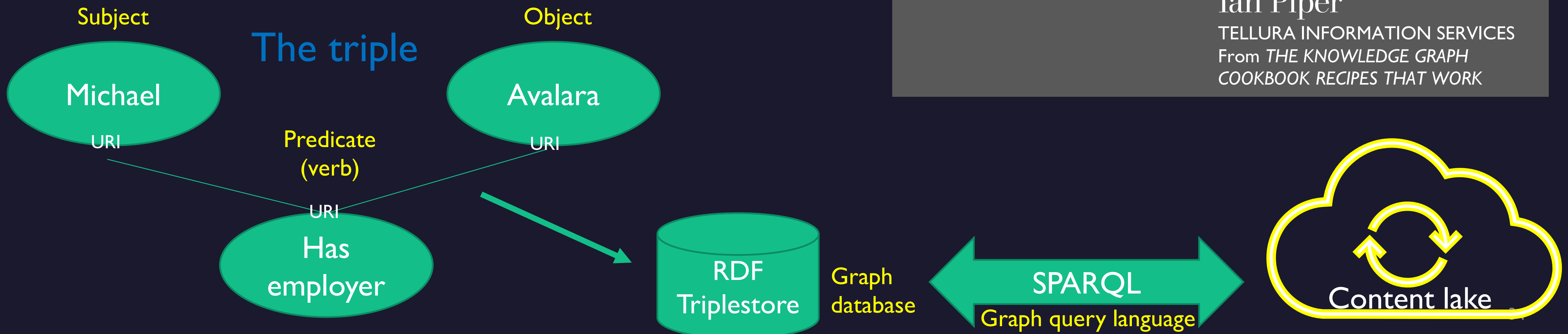
Val Swisher
The Personalization Paradox

Core concept: The triple

- A triple is the core element of structured knowledge bases - ontologies and knowledge graphs. Both ontologies and knowledge graphs are based on "nodes"
- Consists of three basic elements: Subject + Predicate + Object
- *A triplestore* stores these models as a network of objects with materialized links between them

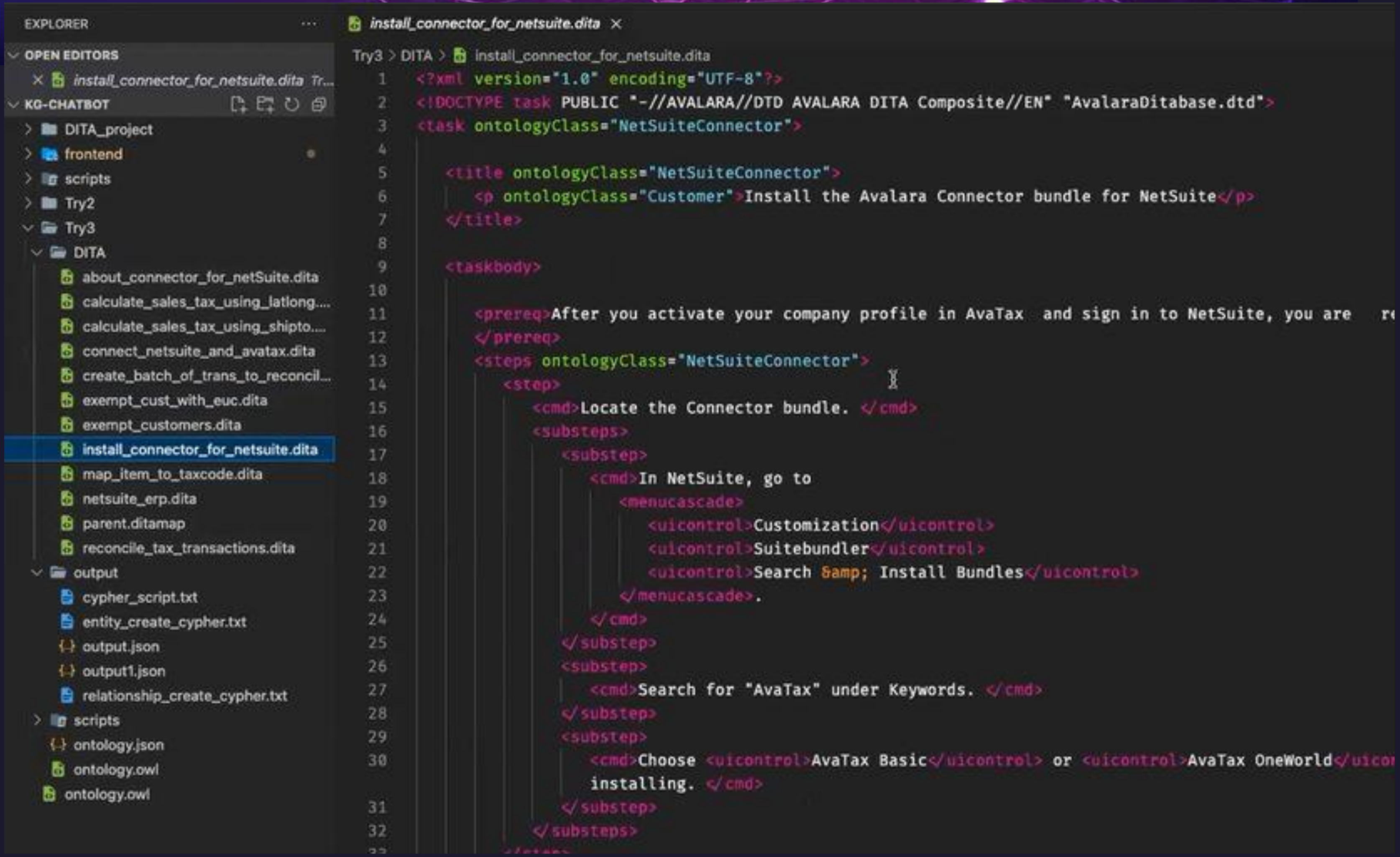
The triple — it is easy to build out massive networks of connected information. This structure then allows sophisticated exploration across this network and offers new insights into the organization's information. [cool video link]

Ian Piper
TELLURA INFORMATION SERVICES
From *THE KNOWLEDGE GRAPH COOKBOOK RECIPES THAT WORK*



What if we tag sub-elements in our structured XML content?

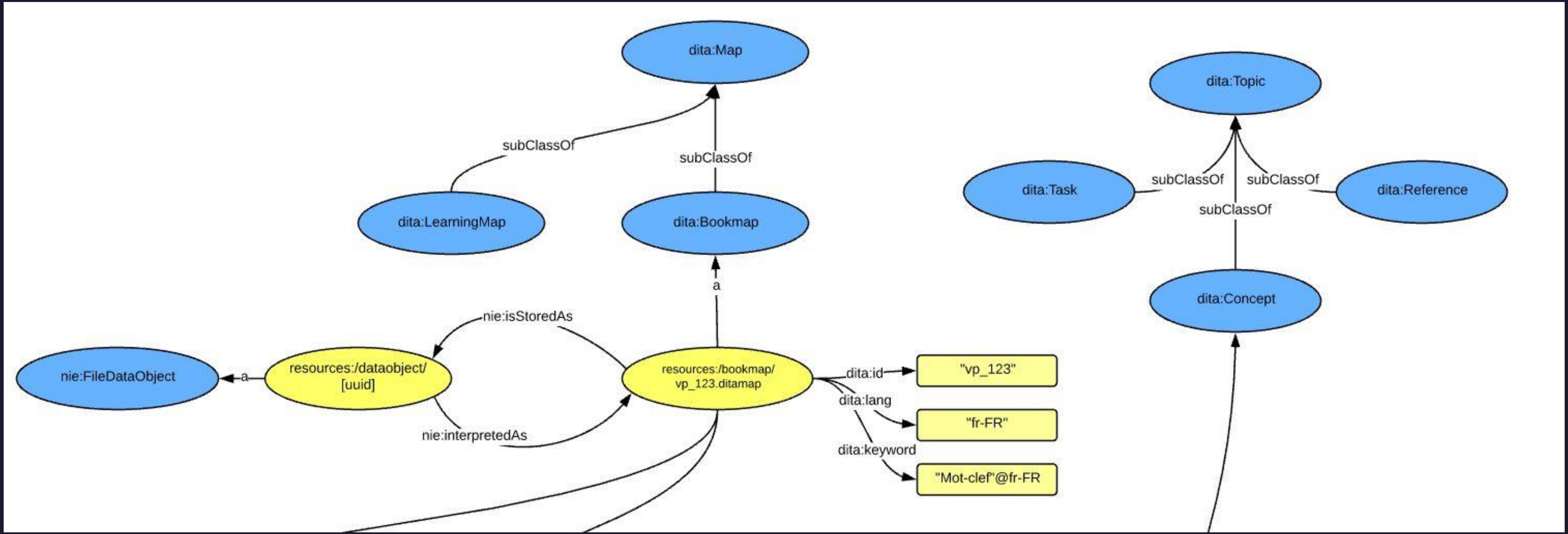
Suddenly we can use our componentized content in new ways to control the tokenization (chunking) normally done by the LLMs.



EXPLORER
OPEN EDITORS
install_connector_for_netsuite.dita Tr...
KG-CHATBOT
DITA_project
frontend
scripts
Try2
Try3
DITA
about_connector_for_netSuite.dita
calculate_sales_tax_using_latlong....
calculate_sales_tax_using_shipto....
connect_netsuite_and_avatax.dita
create_batch_of_trans_to_reconcil...
exempt_cust_with_euc.dita
exempt_customers.dita
install_connector_for_netsuite.dita
map_item_to_taxcode.dita
netsuite_erp.dita
parent.ditamap
reconcile_tax_transactions.dita
output
cypher_script.txt
entity_create_cypher.txt
output.json
output1.json
relationship_create_cypher.txt
scripts
ontology.json
ontology.owl
ontology.owl
install_connector_for_netsuite.dita
Try3 > DITA > install_connector_for_netsuite.dita

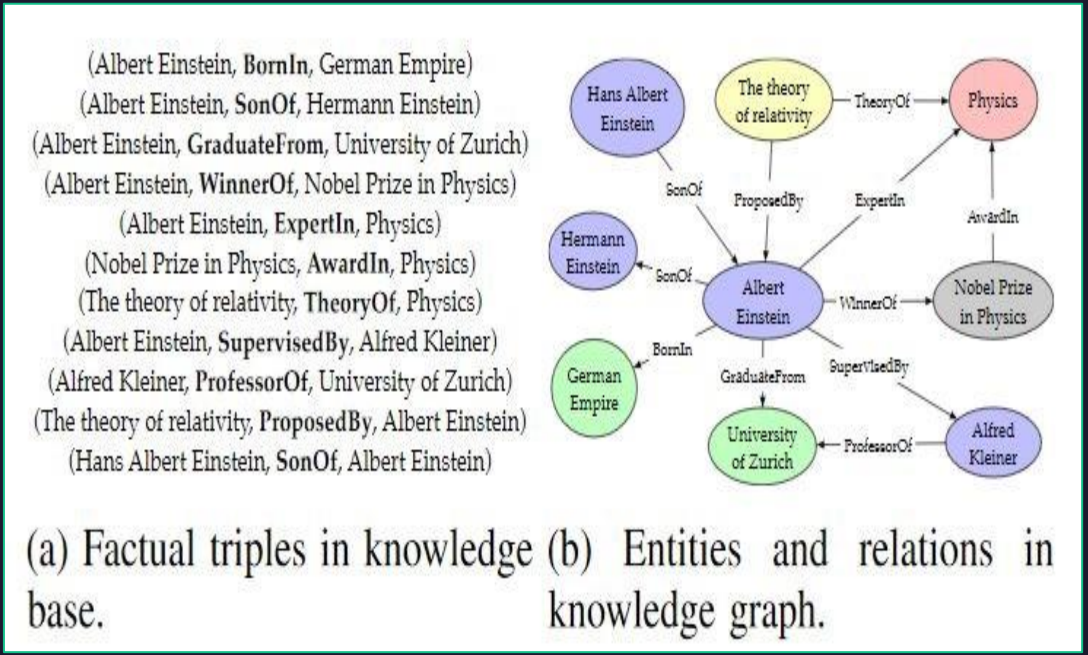
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC "-//AVALARA//DTD AVALARA DITA Composite//EN" "AvalaraDitabase.dtd">
<task ontologyClass="NetSuiteConnector">

    <title ontologyClass="NetSuiteConnector">
        <p ontologyClass="Customer">Install the Avalara Connector bundle for NetSuite</p>
    </title>

    <taskbody>

        <prereq>After you activate your company profile in AvaTax  and sign in to NetSuite, you are   re
        </prereq>
        <steps ontologyClass="NetSuiteConnector">
            <step>
                <cmd>Locate the Connector bundle. </cmd>
                <substeps>
                    <substep>
                        <cmd>In NetSuite, go to
                            <menucascade>
                                <uicontrol>Customization</uicontrol>
                                <uicontrol>Suitebundler</uicontrol>
                                <uicontrol>Search &amp; Install Bundles</uicontrol>
                            </menucascade>.
                        </cmd>
                    </substep>
                    <substep>
                        <cmd>Search for "AvaTax" under Keywords. </cmd>
                    </substep>
                    <substep>
                        <cmd>Choose <uicontrol>AvaTax Basic</uicontrol> or <uicontrol>AvaTax OneWorld</uicon
                        installing. </cmd>
                    </substep>
                </substeps>
```

A sample content ontology modeled on the DITA XML content model



A deeper dive into knowledge graphs

- A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions captured as a semantic graph.
- Where an ontology is a representation a conceptual classes and their relationships, a knowledge graph layers on real object references (via URIs) to an ontology. The graph can be mined to do inferencing and make predictions and improves over time.
- The knowledge graph provides *context* and additional edge knowledge from it that draws those inferences.
- Whereas an ontology specifies the formal semantics of the data, a knowledge graph captures additional intelligence over the stored data for intelligent content retrieval, organization, and delivery.



(Albert Einstein, **BornIn**, German Empire)
(Albert Einstein, **SonOf**, Hermann Einstein)
(Albert Einstein, **GraduateFrom**, University of Zurich)
(Albert Einstein, **WinnerOf**, Nobel Prize in Physics)
(Albert Einstein, **ExpertIn**, Physics)
(Nobel Prize in Physics, **AwardIn**, Physics)
(The theory of relativity, **TheoryOf**, Physics)
(Albert Einstein, **SupervisedBy**, Alfred Kleiner)
(Alfred Kleiner, **ProfessorOf**, University of Zurich)
(The theory of relativity, **ProposedBy**, Albert Einstein)
(Hans Albert Einstein, **SonOf**, Albert Einstein)

(a) Factual triples in knowledge base.

(b) Entities and relations in knowledge graph.

An example of knowledge base and knowledge graph

[A Survey on Knowledge Graphs: Representation, Acquisition and Applications]