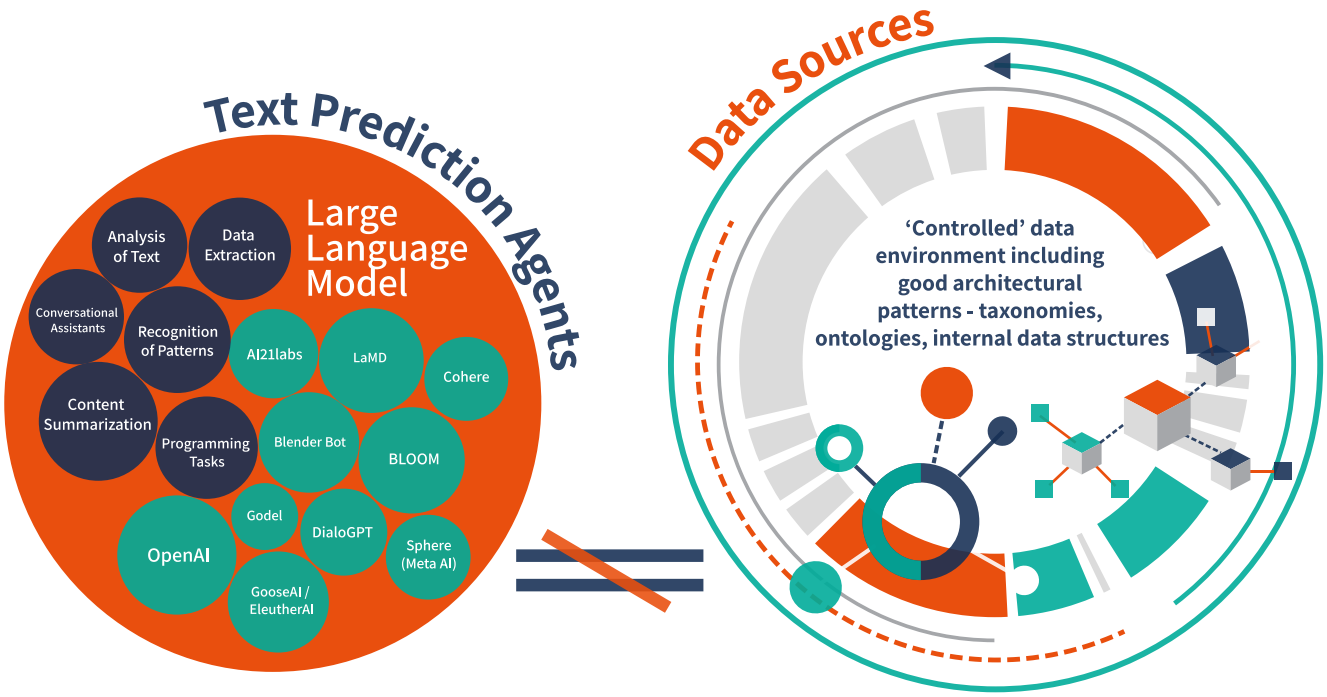
I did some research because I wanted to create a basic framework on the intersection between large language models (LLM) and data management. I will start by saying that I believe LLM holds great promise. But there are also a host of other issues (and cautions) to take into consideration. The technology is very new and not well understood. Most applications are still exploratory. Very few are in production – and this topic has gone from research to applications very quickly.

LLM is by its very design a language model. As such, the logical applications are those associated with the analysis of text and the recognition of patterns. Examples of these types of applications are content summarization, programming tasks, data extraction, and conversational assistants (chatbots). As the technology matures, I believe some of the more valuable applications will be those related to lineage and provenance like using the LLM to facilitate the development of the connected inventory and enhance digital twin capabilities.



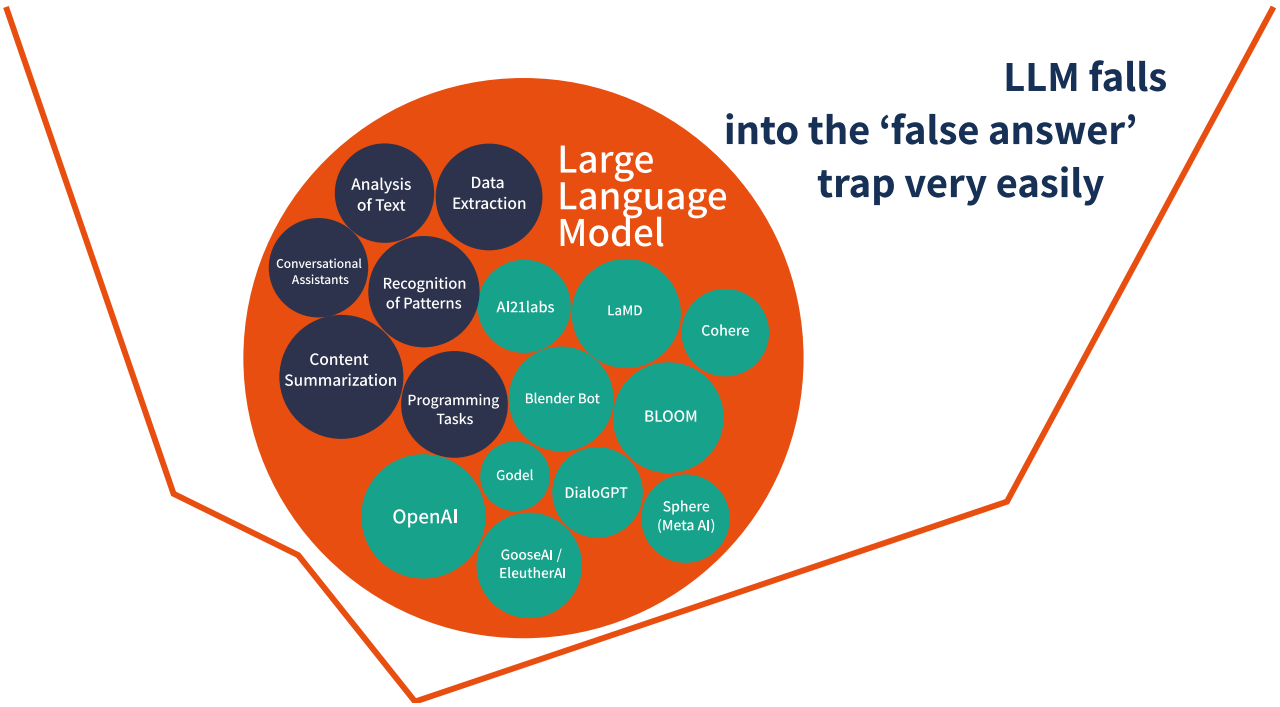
Data Meaning is Critical

It is important to note that LLMs are just 'text prediction agents.' That means they are highly

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

dependent on the quality of the knowledge base (data) being used as input. I urge early adopters to think of this as an extension of their existing efforts to get the data and associated processes within your organization defined, managed, and governed.

For companies considering how to leverage the technology, I suggest the best way to proceed would be to start with a well-executed and 'controlled' data environment including good architectural patterns (i.e., taxonomies, ontologies, internal data structures). Implementing this core data management capability (ideally using semantic standards) should be viewed as a prerequisite for taking full advantage of LLM capability. The meaning of the data is the most important component – as the data models are on their way to becoming a commodity. It was emphasized many times that LLMs are only as good as the data sources.



A Few Cautions

LLM references a huge amount of data to become truly functional, making it a quite expensive and time consuming effort to train the model. Supercomputers (and other

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

components of infrastructure) along with new approaches to data architecture (with billions of parameters) are needed.

Another concern relates to the definition of 'data constraints.' Connecting to a pre-trained LLM is based on APIs. But the interfaces are simplistic – and users must consider how to enforce data policy on shared LLM. Caution is advised on giving LLM access to all internal APIs for security reasons. Managing access and entitlement rights (and where they are stored) is critical and can be problematic.

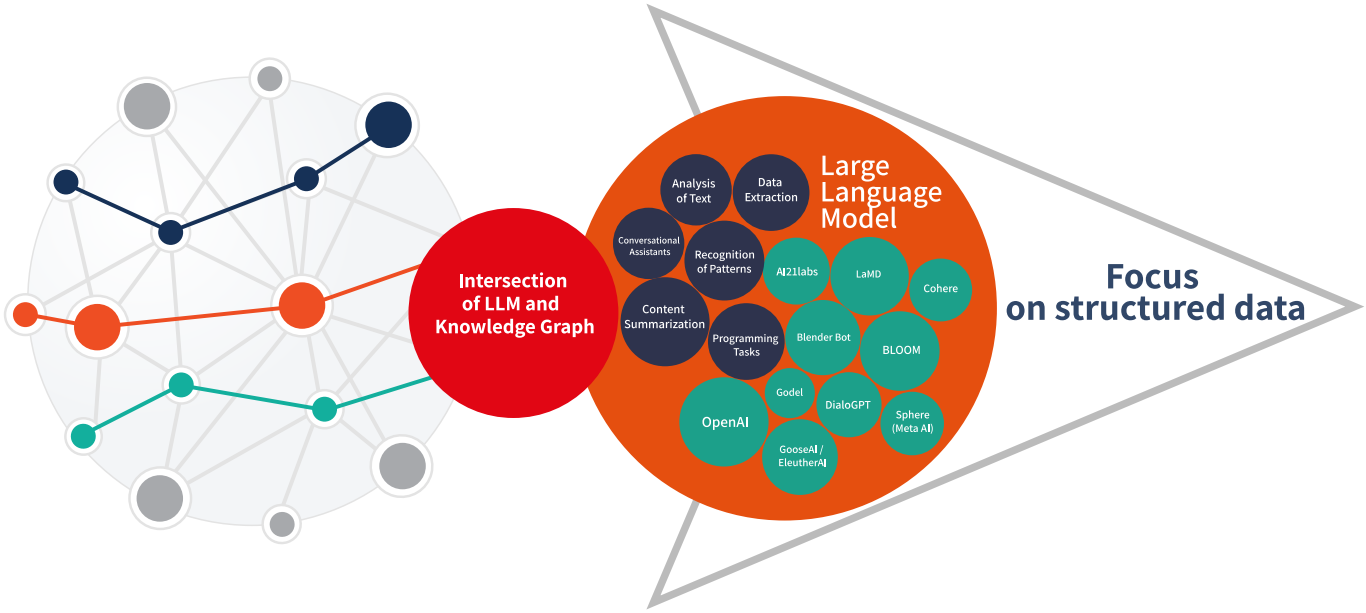
To take advantage of existing LLM, you must agree to make your data publicly available. In that environment, all secrets will be exposed, and all operational aspects must be considered. All data must go through one central place and that can be risky, expensive, and hard to secure. Some companies are banning their use due to privacy concerns.

Quality Concerns

Most LLM have been trained on data from across the Internet which means the quality can be subject to bias. On the quality side, LLM falls into the 'false answer' trap very easily. They take in words without meaning. LLM presents all things as if they were true. If there is duplicate (conflicting) information, the LLM doesn't care – it simply makes up the answer based on the information it has available. Nothing should be considered definitive until it is fact checked.

Training LLM requires building robust data pipelines that are both highly optimized and flexible enough to easily include new sources of both public and proprietary data. Running the LLM on your own cloud environment provides more safety and greater contextual accuracy, but there are very few labs that can train LLM and self-training using your own rules is possible although expensive.

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com



Start with Structured Data

The ideal way to experiment with LLM functionality is to focus on structured data at the start. Cleaning, refining, and aligning your data to shared meaning is the right strategic approach. The LLM can learn from structured content, which allows you to capitalize on content reuse in other areas while continuing to add depth and breadth to the knowledge base.

In other words, get the basics right – focus on areas where you have achieved identity resolution via IRI, meaning resolution via a structured ontology and executable business rules using SHACL. Define your use cases up front and build your ontology based on those use cases. Invest in a ‘prompt engineer’ that can link the LLM to other data via your organization’s APIs. Prompting is an iterative process and is dependent on the models the LLM is trained on. LLM can’t learn without the ontology. They can’t protect your sensitive data without the rules. And they can’t guarantee quality without oversight from a properly managed Office of Data Management.

It might be difficult to resist the temptation to jump right into the LLM pool – but remember

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

ontotext

that LLM is still maturing as a self-service capability. And while there is a high expectation that LLM will mature rapidly – most activities and applications are still in their experimental phase. I am personally excited about the intersection of LLM and semantic standards (knowledge graph). The prerequisite for using LLM most effectively is to get your data architecture and engineering. Experiment away – but be careful you don't get burned at the top of the hype curve.

Want to learn more about the intersection of LLM and knowledge graphs?

Book a Discovery Call

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com