

The promise of Large Language Models

A complete new industry/ecosystem of innovative applications on top of LLMs

Promise – automate human tasks

- **Hide technology complexity – (SPARQL) query generation on (large) distributed data sets**
- **Automatic transformation of data sets**

Issues with LLMs (like chatGPT)

- **Precision – it works or doesn't**
- **Hallucinations – how to distinguish incorrect ('fake') from correct results**
- **Short tail – works for general problems very well, not for specific ones**
- **Data freshness – how actual is the training set? Is it trained with the FEDeRATED ontology and data sets (e.g. chatGPT3.5 is trained with upto 2021 data)?**
- **(Energy consumption – the (large) amount of computational power)**



Ontology chatbot – natural language interaction with semantic data



Are we able to generate (SPARQL) queries? What type of expertise is required (technical, logistics, semantic model)?

- **Solutions**

- **Stardog – Voicebox tutor (prototype)**
- **Voicebox is based on Langchain python package (open source)**
- **Langchain has a natural text to SPARQL generator (GraphSparqlQChain)**
- **Ontotext – integration of chatGPT in GraphDB as (SPARQL) chatbot**
- **Small experiment (7 users) – promising, but requires further work**
 - **Increased trustworthiness of the tools (improved GUI)**
 - **Improve user interactions**
 - **Better support of the ontology (FEDeRATED/logistics specific tool?)**



Ontology alignment and matching



Is it possible to do on the fly data transformation with an intermediate ontology?

- **Alignment**
 - other tools can be applied (prototype Service Registry)
 - There is a fine-tuned LLM part (BERTMap) part of another tool (DeepOnto)
- (Traditional) matching is based on text mining, NLP (Natural Language Processing) and neural networks
 - Does not meet the required precision, correctness, and correctness
- General purpose LLMs for initial matchings (chatbot) → fine tuning and testing by human
- A requirement for fine-tuned LLMs → is foreseen to come!

