Introduction to Large Language Models (LLMs)

Prof. Tanmoy Chakraborty, Prof. Soumen Chakraborti
Department of Computer Science & Engineering
Indian Institute of Technology, Delhi

Lecture 26

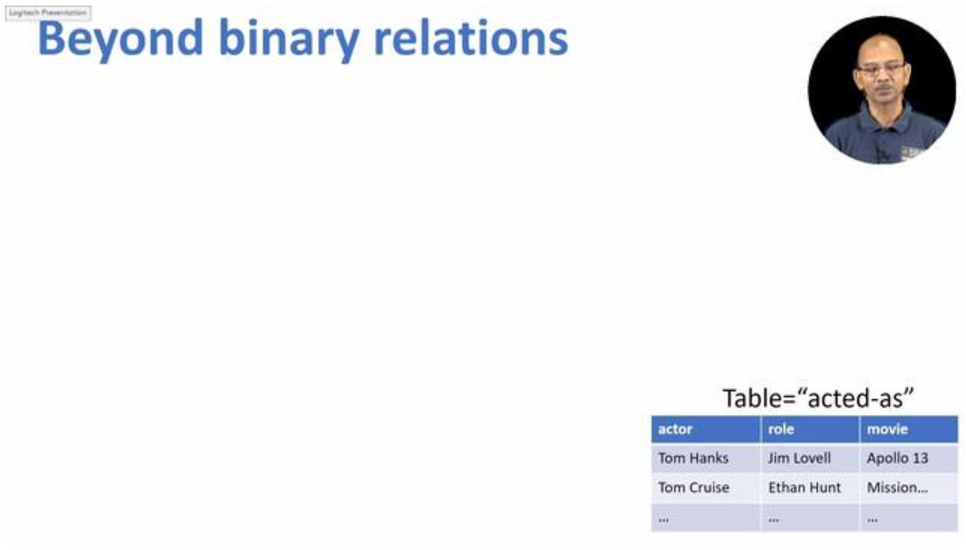
Knowledge and Retrieval: Knowledge Graph

Hi, welcome back to the course on LLMs. In this module, we will cover knowledge graphs and their retrieval from large language models. From the ancient times, the Rosetta Stone the library in Alexandria to Vannevar Bush's vision of the Memex machine, humans have tried to organize knowledge in one form or another. In recent times, in computers, that has taken the form of knowledge graphs, such as Yago, WordNet, Freebase, and Wikidata. For our purposes, a knowledge base, or interchangeably a knowledge graph, will have nodes that represent entities such as Barack Obama or Michelle Obama or Honolulu or USA.

Logitech Presentation
1
Dual projectors with the ability to cross reference text (hypertext).
User can leave thier own notes and comments
Rick Astley
Human interfaces with machine through buttons and has the ability to make new hyperlinks and associations (like bookmarks) and automated search and export user content to other memex machines
3
A sewing machine
Sum of all human knowledge stored on micro-film in the form of text / image compositions
2
Global WordNet Association
Cycorp
WIKIDATA
yago select knowledge
Freebase

And these entity nodes will be connected with edges that are labeled with a relation type. For example, born in. So Barack Obama was born in Honolulu, which is a city of USA. A KG triple or fact will therefore look like a subject, a relation and an object. So this chart shows a knowledge graph with binary relations where two entities can participate in a relationship.

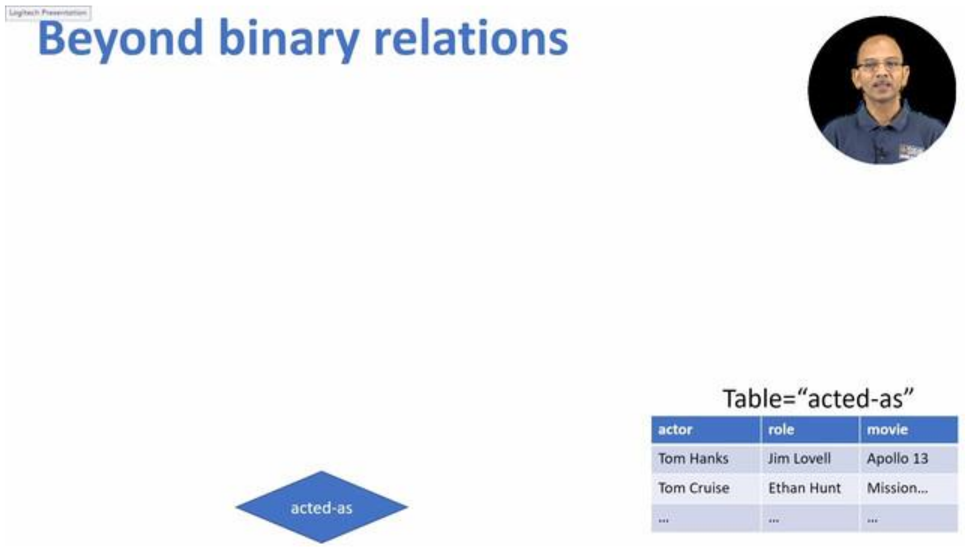
We can, however, go beyond binary relations, although we won't be dwelling on it in that much detail in this course. For example, we may start with a table that has three columns, an actor, his or her role in a movie and the name of the movie. For example, Tom Hanks played the role of Jim Lovell in Apollo 13 or Tom Cruise played the part of Ethan Hunt in Mission Impossible. In a graph form, we will represent this as follows. The acted as table or relationship will be represented by diamond shaped nodes as shown.



Beyond binary relations

Table="acted-as"

actor	role	movie
Tom Hanks	Jim Lovell	Apollo 13
Tom Cruise	Ethan Hunt	Mission...
...	...	...

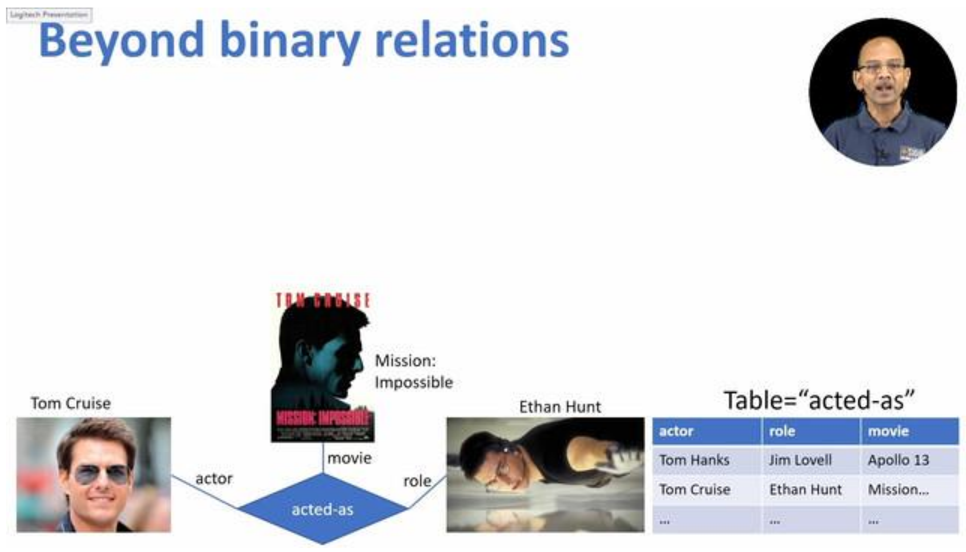


Beyond binary relations

Table="acted-as"

actor	role	movie
Tom Hanks	Jim Lovell	Apollo 13
Tom Cruise	Ethan Hunt	Mission...
...	...	...

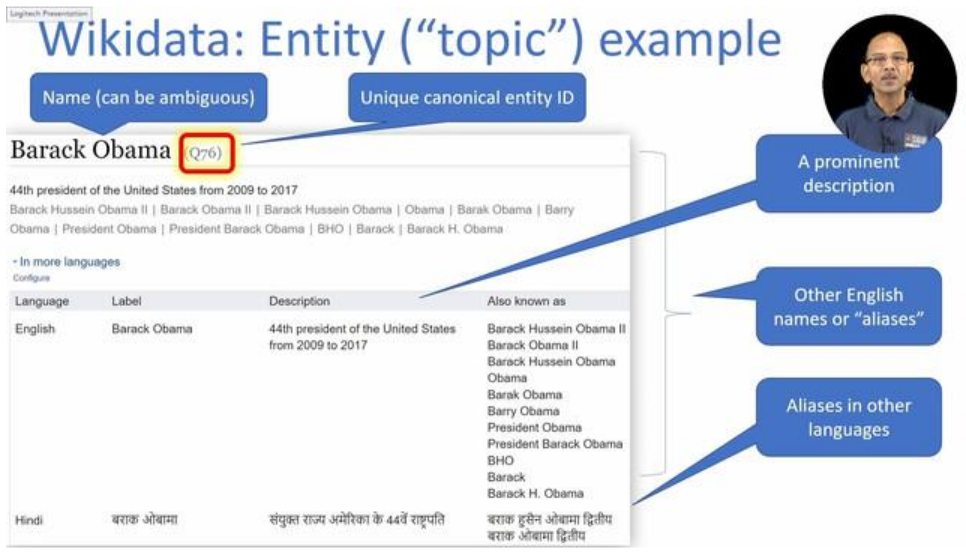
And Tom Cruise and the movie and the role would be connected to this relation. Similarly, if you have president of relationship where Barack Obama was president of USA between 2009 and 2017, that would be shown as a president of node connected to all the other aspects or elements of the relationship. Barack Obama is the subject entity, USA is the object entity, and there is a start year and an end year as attributes. If you go to the largest public knowledge graph today, that is Wikidata, and look for Barack Obama, you will find this particular page. What does it show? It shows the name Barack Obama.



Beyond binary relations

Table="acted-as"

actor	role	movie
Tom Hanks	Jim Lovell	Apollo 13
Tom Cruise	Ethan Hunt	Mission...
...	...	...



Wikidata: Entity ("topic") example

Barack Obama (Q76)

44th president of the United States from 2009 to 2017

Barack Hussein Obama II | Barack Obama II | Barack Hussein Obama | Obama | Barak Obama | Barry Obama | President Obama | President Barack Obama | BHO | Barack | Barack H. Obama

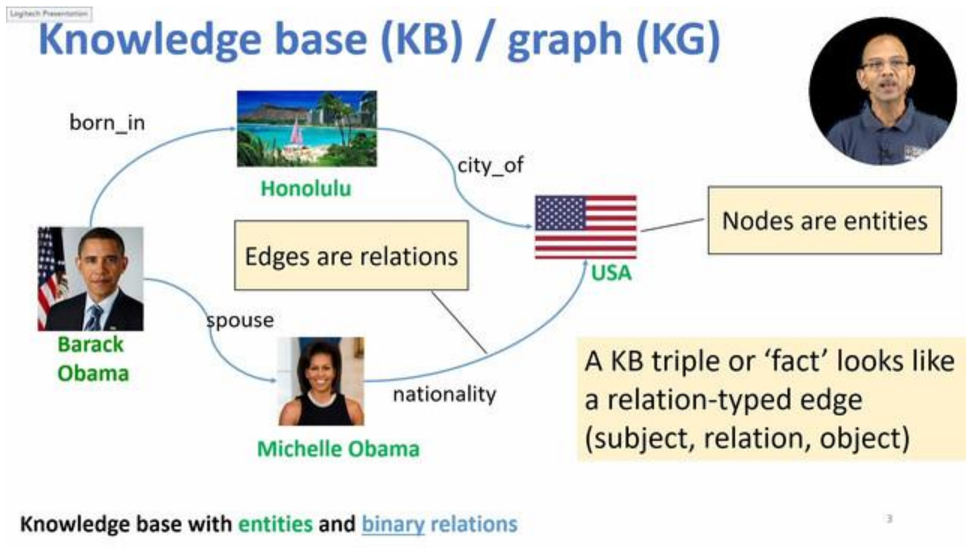
Language	Label	Description	Also known as
English	Barack Obama	44th president of the United States from 2009 to 2017	Barack Hussein Obama II Barack Obama II Barack Hussein Obama Obama Barak Obama Barry Obama President Obama President Barack Obama BHO Barack Barack H. Obama
Hindi	बराक ओबामा	संयुक्त राज्य अमेरिका के 44वें राष्ट्रपति	बराक हुसैन ओबामा द्वितीय बराक ओबामा द्वितीय

However, this can be ambiguous. Barack Obama as a name might be reasonably unique, but John Smith or David Carmel, those names could be much more ambiguous. To make things unambiguous, unique, and canonical, there will be an entity ID. In the case of Wikidata and Barack Obama, that number is Q76. There is no meaning in that string.

It's just a unique string that will not be shared by any two people or any two entities in Wikidata. So that's Barack Obama's unique canonical entity ID. Then there would be a prominent description, 44th President of the United States. So that would more or less disambiguate this person from any other person in the knowledge graph. Then there would be other English names or what are called aliases, Barack Hussein Obama II or II, Just Obama, Barry Obama, BHO, etc.

But Wikidata is also a multilingual knowledge graph. It will have other aliases of the same entity in a multitude of languages of the world. So this covers entities. So this is an example entity in Wikidata. On the relation side, if we scroll down on the previous page of Barack Obama, eventually we'll come to a relation which will be called position held.

And what is the position held? He has held many positions, but the most prominent one being the president of the United States. And this has attributes like start time and who did he replace, how was he elected, what was the end time, who was he replaced by, and so on and so forth. If you now click on position held, you will go to another page of Wikidata, which is dedicated to that relation. Once again, just like Barack Obama, the relation position held will not depend on only this string description, but it will have this opaque unique ID, this time starting with a P instead of a Q, P39, which is the unique ID assigned by Wikidata to this particular relationship. And once again, just like its description in English and its longer explanation, We are going to have its aliases not only in English, but also in many other languages of the world.

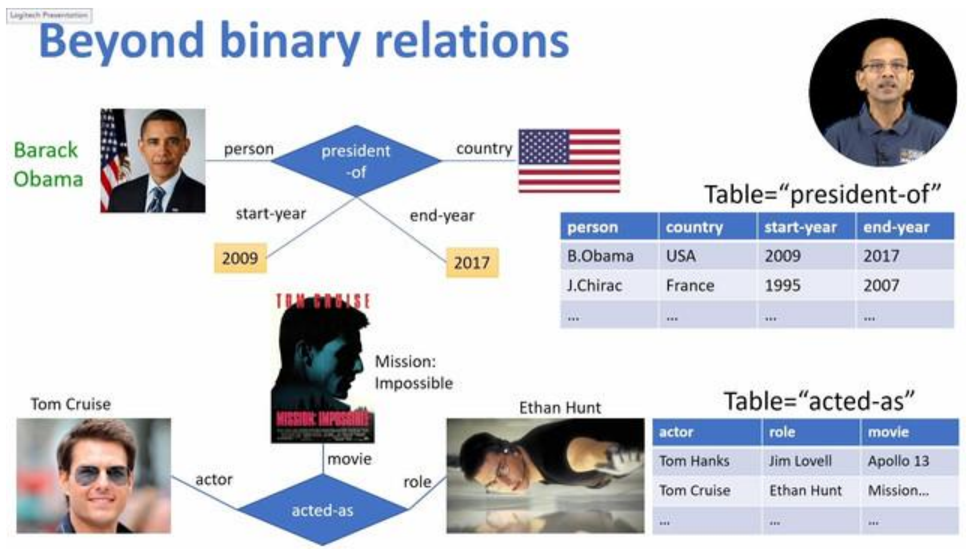


Knowledge base (KB) / graph (KG)

A KB triple or 'fact' looks like a relation-typed edge (subject, relation, object)

Knowledge base with entities and binary relations

Here I have shown two Indian languages. So that more or less defines a knowledge graph as far as we shall need it. Now, knowledge graphs are extremely useful. They record authoritative, structured information from the world, which we can use for answering questions and for search. However, because human inspection is involved in entering facts into a knowledge graph, facts are added relatively slowly.



Table="president-of"

person	country	start-year	end-year
B.Obama	USA	2009	2017
J.Chirac	France	1995	2007
...	...	...	...

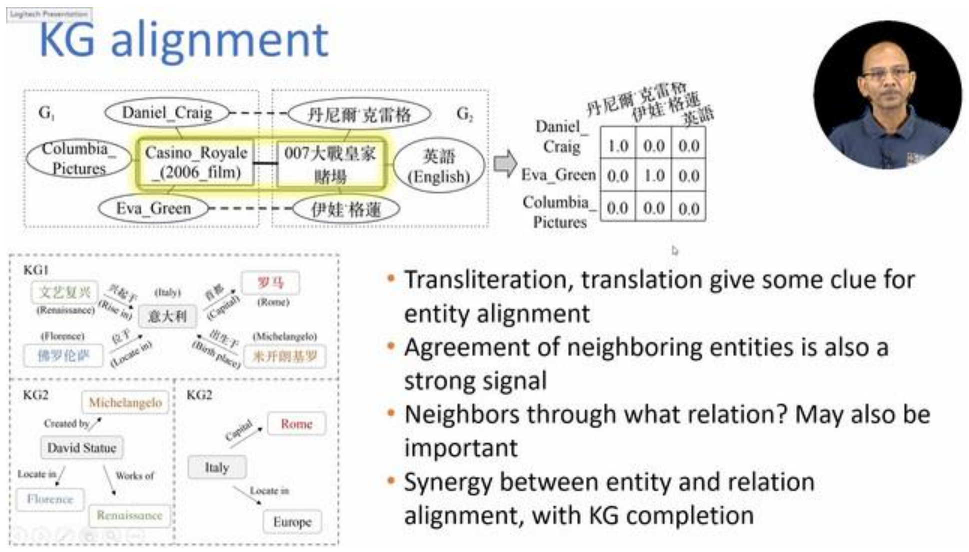
Table="acted-as"

actor	role	movie
Tom Hanks	Jim Lovell	Apollo 13
Tom Cruise	Ethan Hunt	Mission...
...	...	...

And therefore, most knowledge graphs are useful, but they're very incomplete. For example, Freebase, another big knowledge graph, has the place of birth unknown for 93.8% of people inside Freebase. This makes knowledge graph or knowledge base completion a very important task. We need to complete a knowledge graph not only explicitly so that the knowledge graph can grow, but also implicitly when they are consulted by large language models and question answering systems.

We shall see this when later on in this module we reconcile knowledge graphs with large language models. Another important task around knowledge graphs is alignment between knowledge graphs in different languages. Here we show fragments of an English knowledge graph and the Chinese knowledge graph. As you can see, translation and transliteration can give certain clues about which entities in English should align with which entities in Chinese. However, for a full informed judgment of entity to entity alignment, it also helps to note what their neighboring entities are and how they are connected with what relationship they are connected.

And that is also shown in these simplified diagrams. Now, as you can see, these two nodes, CasinoRoyal and its counterpart in Chinese, will line up, if we can also establish that their neighbors in turn are lined up. There turns out to be a synergy between entity alignment and relation alignment and cage completion. So all these tasks are actually interrelated with each other. As I was saying, a very important application of knowledge graphs is in question answering.



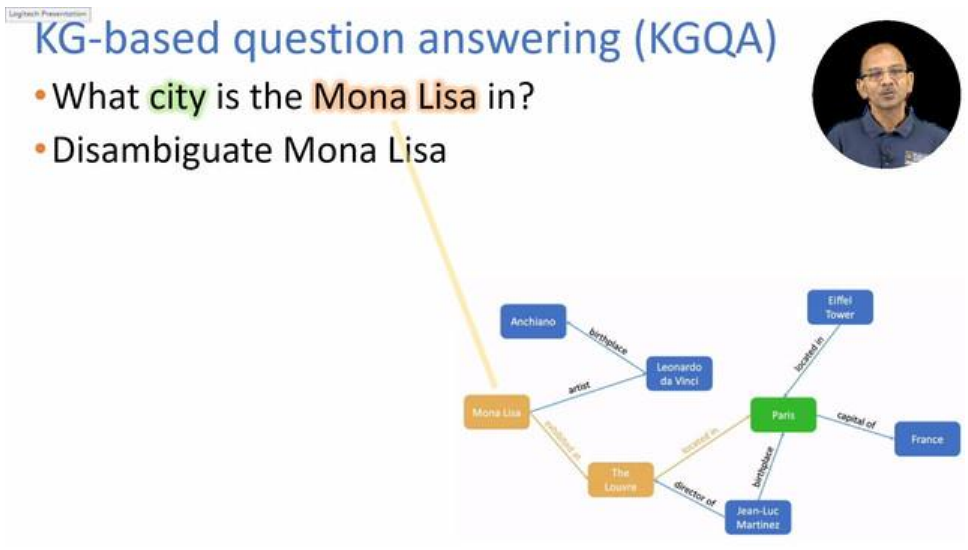
KG alignment

	丹尼爾·克雷格	伊娃·格蓮	英語
Daniel_Craig	1.0	0.0	0.0
Eva_Green	0.0	1.0	0.0
Columbia_Pictures	0.0	0.0	0.0

- Transliteration, translation give some clue for entity alignment
- Agreement of neighboring entities is also a strong signal
- Neighbors through what relation? May also be important
- Synergy between entity and relation alignment, with KG completion

Suppose we have this fragment of a graph shown below, where we have Leonardo da Vinci, who was an artist who created the Mona Lisa. The Mona Lisa is exhibited at the Louvre Museum, which is located in Paris, and so is Eiffel Tower, and Paris is the capital of France, et cetera. In this setting, if we ask what city is the Mona Lisa in? What is involved? I have to trace a path from Mona Lisa to Paris because Paris is a capital and therefore it's a city. Right. And the other elements around may provide evidence like the Louvre provides a connective evidence between Mona Lisa and Paris.

But the Louvre itself is not involved in the answer. So first, we have to disambiguate Mona Lisa. Maybe there are other entities called Mona Lisa. And based on the query, we may have to use a lot of word knowledge to infer what the most likely connotation of Mona Lisa is. And large language models are excellent at this task.



KG-based question answering (KGQA)

- What city is the Mona Lisa in?
- Disambiguate Mona Lisa

They have collected word knowledge from millions and billions of utterances and sentences in a large web corpus. So it finds it very easy to understand that the dominant meaning of Mona Lisa is the portrait. So first we have to design Mona Lisa. And then we have to infer that if a painting is exhibited at a particular museum and that museum is located in a particular city or location, then Mona Lisa would also be located in that city. Now observe that there are two different parts of the reasoning.

One is the factual world knowledge about where Mona Lisa is located. Tomorrow, in principle, although it's very unlikely to happen, France may gift the Mona Lisa painting to another nation like Japan or Germany. And then the position of Mona Lisa would change and the knowledge graph can easily be changed to reflect that modification. Wikipedia, another knowledge graph plus structured corpus, is carefully curated by experts in every field. And when typically a significant development happens in any field, within minutes or within hours, the corresponding Wikipedia pages are updated.

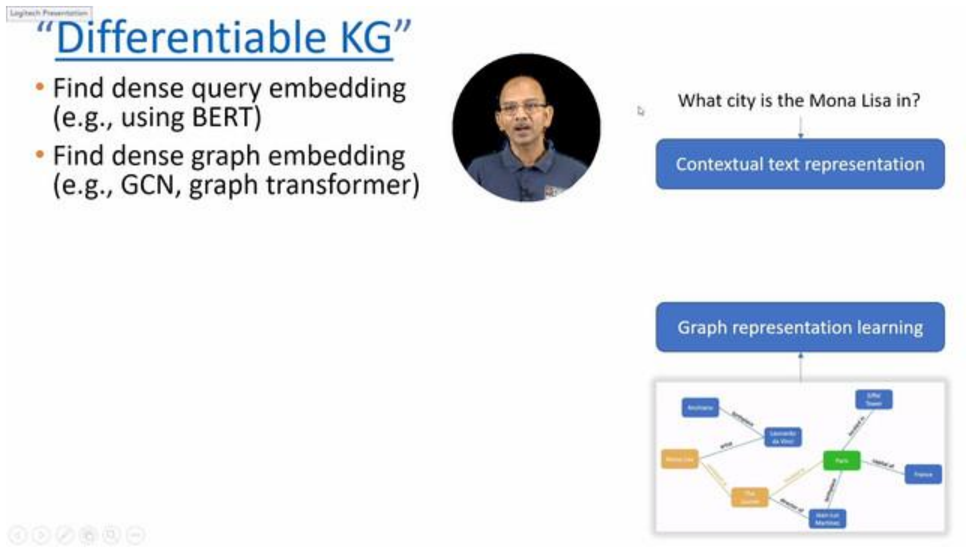
A large language model, in contrast, cannot be updated that fast because it has a humongous number of model parameters and takes a lot of computation to train and retrain. So that's the first part that we need to consult world knowledge to keep track of possibly changing data. The second part, however, is this inference of what Mona Lisa might mean and the word knowledge that a painting in a museum in a city means the painting is in the city. And for this, we might use the reasoning powers of large language models. So both these capabilities have to come together to solve this problem.

And we also have to look up Paris to find out that it's a city. That's actually quite easy. And so there are two approaches to solving these kinds of problems. One is called semantic interpretation, where the original textual query will be converted into a structured query language, such as SQL or SPARQL, and then it is executed on a structured database representing the graph data. I'll talk less about that in this module, and I'll talk more about the so-called end-to-end trainable knowledge graphs.

Still, to make things complete, let's discuss semantic interpretation for a minute. So as I was saying, you would like to translate the natural language question to a structured query that is cognizant of the KG schema. And then we want to execute the structured query on the cage. So if we start with what city is the Mona Lisa in, after semantic parsing, we'll end up with this fairly complicated looking query. Select city where museum is located in city and Mona Lisa is exhibited at museum.

Will this work for our specific knowledge graph? That depends on the schema. If Mona Lisa itself is mapped to a unique painting ID, then I need another hop in this process to force disambiguate Mona Lisa to an ID, and then try to find out in which museum it is exhibited. After that, we also need to do this kind of entity resolution where Mona Lisa is disambiguated to this entity Q12418, and then that is executed on the knowledge graph. This makes the process very interpretable to experts. An expert who understands SPARQL can look at the query with its embedded constants and find out if the translation was correct or not.

But it is quite tricky to implement, and it's not so easily end-to-end trainable. For this reason, we will spend more of our time on the so-called differentiable knowledge graph abstraction. So once again, here I directly have the graph as we showed before. And we start with the query, what city is the Mona Lisa in? First, we use a dense embedding for the whole query. As we have seen in earlier parts of this course, there are encoders like BERT or BART which can turn this whole query into one or more dense vectors. At that point, I have to also look at the knowledge graph itself and turn it into a compatible neural representation. So on both sides, I now end up with dense embeddings, a dense query embedding and a dense graph embedding. The dense graph embedding may be arrived at using a graph convolutional network, a graph neural network, or a graph transformer, or any of the techniques we are about to explore in this module. Once both sides have a dense representation, we can have a cross attention network in some ways similar to the kind of attention we have been discussing in language models to identify the answer.

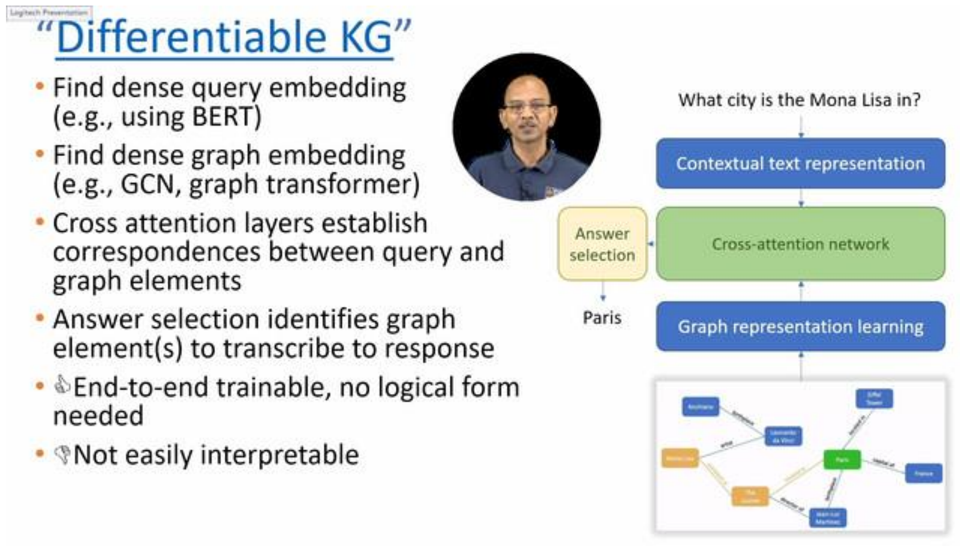


"Differentiable KG"

- Find dense query embedding (e.g., using BERT)
- Find dense graph embedding (e.g., GCN, graph transformer)

What city is the Mona Lisa in?
Contextual text representation
Graph representation learning

So the answer selection will identify a subgraph or graph elements that represents the desirable response. The advantage in this kind of paradigm of question answering is that it is end-to-end trainable and we don't need a logical form or a formal language to express the query. The negative side is that this is not easily interpretable. This again goes back to a large language model's black box feeling. So when these systems misbehave, they are somewhat harder to debug.



"Differentiable KG"

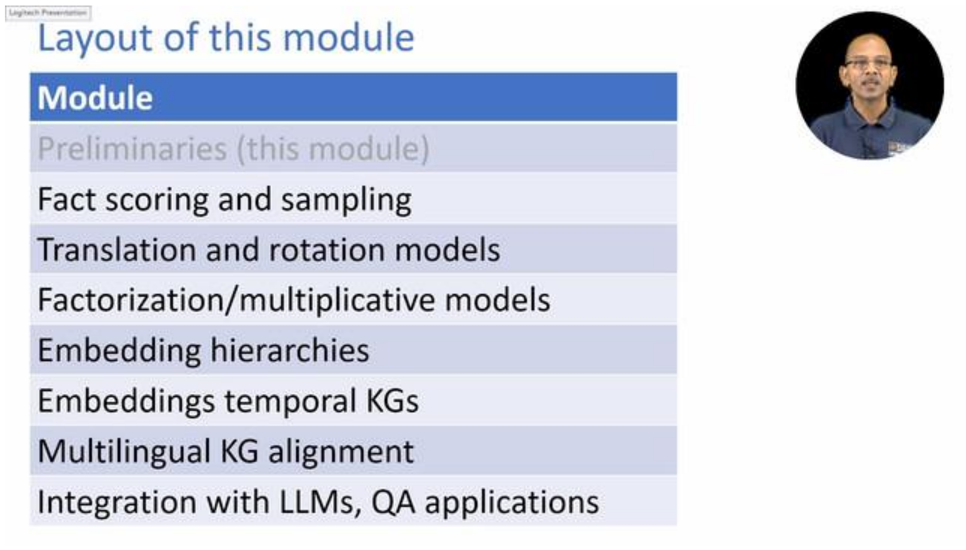
- Find dense query embedding (e.g., using BERT)
- Find dense graph embedding (e.g., GCN, graph transformer)
- Cross attention layers establish correspondences between query and graph elements
- Answer selection identifies graph element(s) to transcribe to response
- 👍End-to-end trainable, no logical form needed
- 👎Not easily interpretable

What city is the Mona Lisa in?
Contextual text representation
Cross-attention network
Answer selection
Paris
Graph representation learning
Leonardo da Vinci
Paris
France

Why do we want to put LLMs and KGs together? I've already given some of the reasons in earlier parts of this lecture. Presidents, ministers, CEOs, champion teams, they all keep changing. Today's LLMs embed these kinds of factual knowledge in their opaque, vast parameter space, and they're very costly to retrain. Aesthetically, it's much better to isolate this kind of structured mutable knowledge from the language understanding and reasoning abilities that LLMs show. This isolation can also guard against hallucination.

In other parts of this course, we have discussed or we will discuss the tendency of large language models to make up fictitious things through their token sampling process. Consulting a knowledge base either by itself or after probabilistic completion is a great way to prevent the LLM from hallucinating. It also makes LLMs more interpretable. Suppose we build a question answering system which is only using an LLM and nothing else. When that misbehaves, it's very hard to go debugging that giant transformer network to find what went wrong.

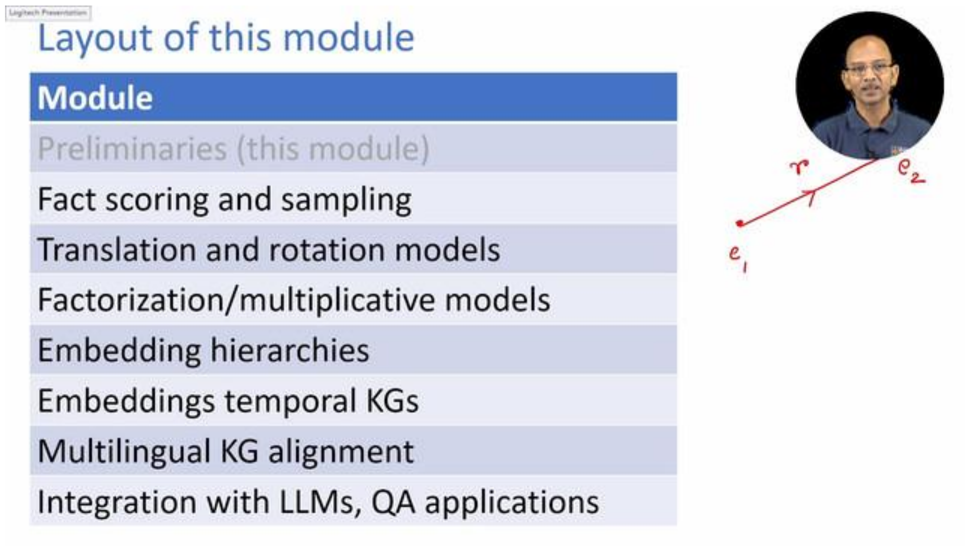
However, if we look at the LLM's attention pattern on the knowledge graph and which regions of the knowledge graph it consults or touches, it may become much more easy to decide how that particular query went wrong. Here is the layout of this module. I have just covered the preliminaries so far. In the next module, we will talk about how facts in the knowledge graph are scored and sampled. After that, we will start with some of the most successful knowledge graph representation paradigms, for example, translation and rotation models.



Layout of this module

Module
Preliminaries (this module)
Fact scoring and sampling
Translation and rotation models
Factorization/multiplicative models
Embedding hierarchies
Embeddings temporal KGs
Multilingual KG alignment
Integration with LLMs, QA applications

In the translation model, we will have entities as points. And one entity, the subject, will be mapped to another entity, the object, through a displacement which corresponds to the relation. That's the translation model. On the other hand, in the rotation model, we will have one entity as a vector. The relation will rotate it to the other entity, which is the tail or the object of the relationship.



Layout of this module

Module
Preliminaries (this module)
Fact scoring and sampling
Translation and rotation models
Factorization/multiplicative models
Embedding hierarchies
Embeddings temporal KGs
Multilingual KG alignment
Integration with LLMs, QA applications

These are very interesting techniques, but they don't achieve the best possible performance, which is achieved by factorization or multiplicative methods. The idea there is that I have a tensor or a three-dimensional matrix which is a factorization between the subject the object and the relation space we want to represent this gigantic tensor with a much simpler factorized form and that is the factorization or the multiplicative model Then we will discuss certain limitations of both these classes of models. For example, representing hierarchies. A camel is a mammal is an animal. It's not clear that any of these methods of representation can handle hierarchies very well.

On the other hand, question answering is full of exploiting hierarchies. I might ask a question like which scientist played the violin? And my answer would be Einstein, maybe among other scientists. But here I had to use that critical information that Einstein is a scientist. Sometimes the knowledge graph will explicitly tell you that Einstein is a scientist. For someone as famous as him, that will of course happen.

But there may be lesser known entities whose types may not all be registered. In fact, even if you look at Barack Obama, his prominent type is of course that he is president. But he's also a father, he's also a husband, he's also a book author. And not all these types may be registered in this incomplete knowledge graph. Then we'll spend a little time on the issue of time in knowledge graphs.

As we saw, Barack Obama is president of the US only in a limited time range. Suppose I start asking questions where time is a very important variable. For example, who was the prime minister of Japan when the Schumacher-Levy comet smashed into Jupiter? So in this case, I have to first find out when the Shoemaker-Levy comet smashed into Jupiter, and then consult the knowledge graph to find out who was prime minister of Japan at that time. Now, if you take these examples to ChatGPT today, you will get the impression that everything we are learning in this module is unnecessary because ChatGPT can do that already. But part of the reason is that all these examples I'm mentioning involve famous people and famous events.

We know through our research that as soon as you drop off the popular or head entities into the torso or less known entities, or maybe entities in less endowed, lower resource languages and cultures, the coverage of knowledge graphs fall quite drastically. And there, the synergy between language models and knowledge graphs needs to really work well to have effective question answering applications. That is the end of the current module.