CS639 Deep Learning for NLP

LLMs and Knowledge Graphs

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Slides adapted from Graham, Zhengbao and the survey from Pan et al.

https://junjiehu.github.io/cs639-spring26

Goal for Today

- Types of Knowledge Graph (KG)
- Integration of LLMs and KG
  - **KG-enhanced LLMs**
  - **LLM-augmented KGs**
  - **Synergized LLMs + KGs**
- Future Research Directions

Recap: Transformer LLMs

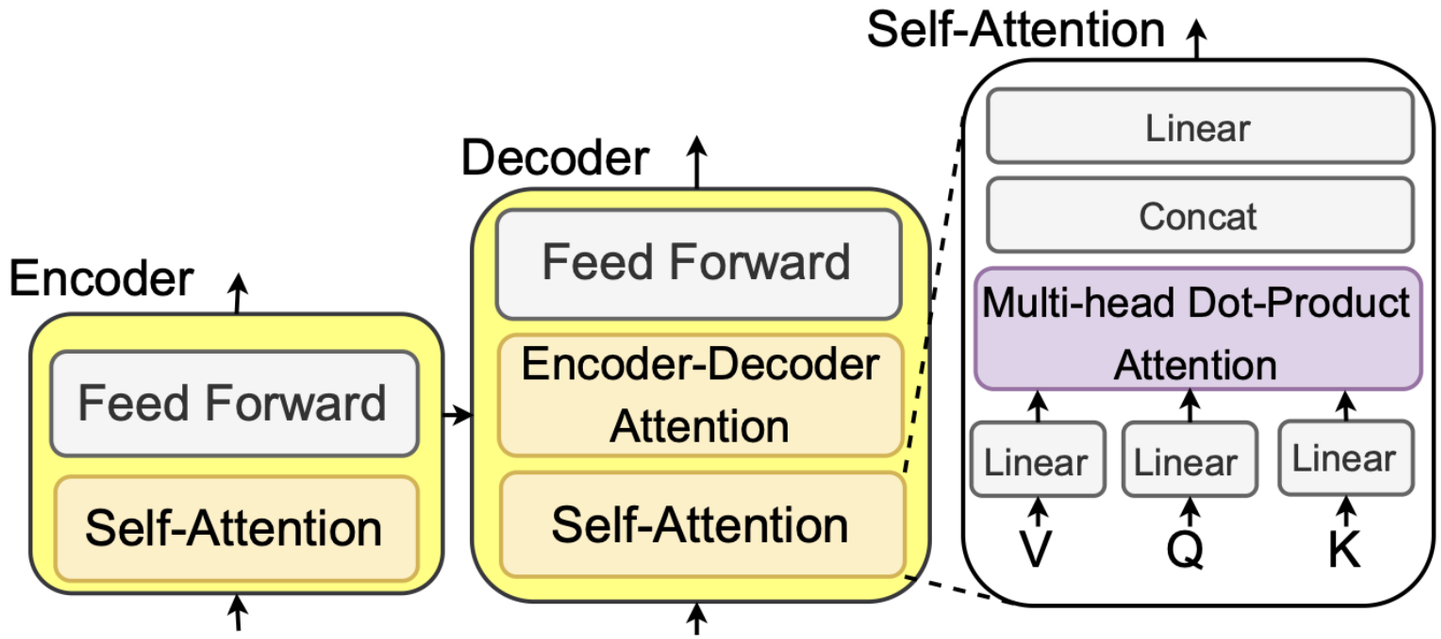
- **An encoder-decoder Transformer** (a.k.a. seq2seq model) consists of
  - A Transformer **encoder**: summarize input contexts
  - Another Transformer **decoder**: attend to input contexts, and iteratively generate words one by one
- **A Transformer layer**: consists of a multi-head self-attention layer, and two feedforward layers.
- **Autoregressive LMs**: generates the next word given the prefix context

$$P(X) = \prod_{i=1}^{I} P(x_i \mid x_1, \ldots, x_{i-1})$$

Next Word

Context

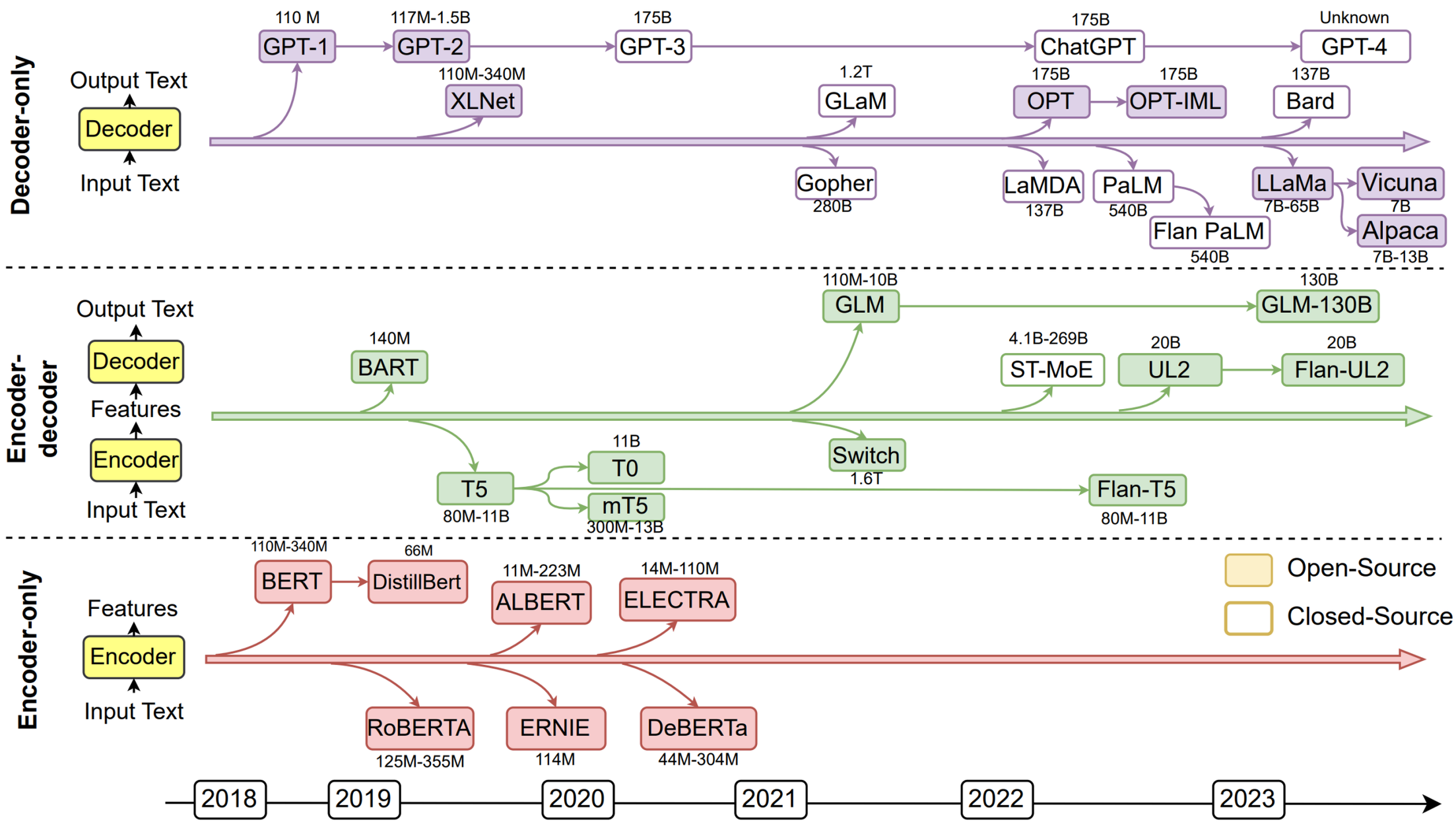
(See LM lecture #5)



(See Transformer lecture #7)

Large Language Models (LLMs)

- Overparameterized NNs pretrained on massive texts by self-supervised objectives



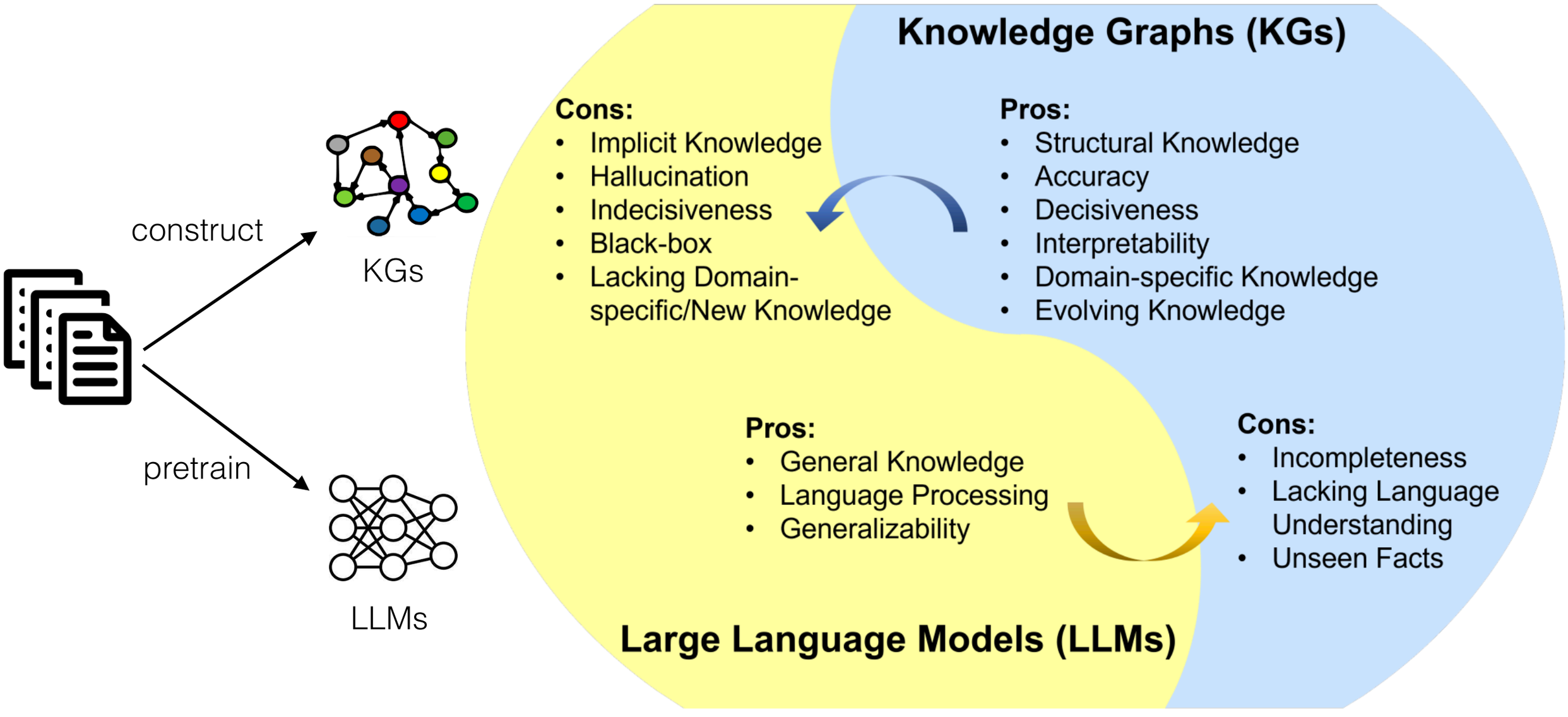
(See Pretraining lecture #8)

Knowledge Graph (a.k.a. Knowledge Base)

- Structured databases of knowledge usually containing
  - Entities (nodes in a graph)
  - Relations (edges between nodes)
- KG Constructions:
  - Manual curation by human experts
  - Information extraction by ML models or heuristics
  - Hybrid human-in-the-loop approach

LLM vs. KG

- Both LLMs and KGs store knowledge in a parametric and non-parametric way



Pan et al. Arxiv 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap
https://arxiv.org/pdf/2306.08302.pdf

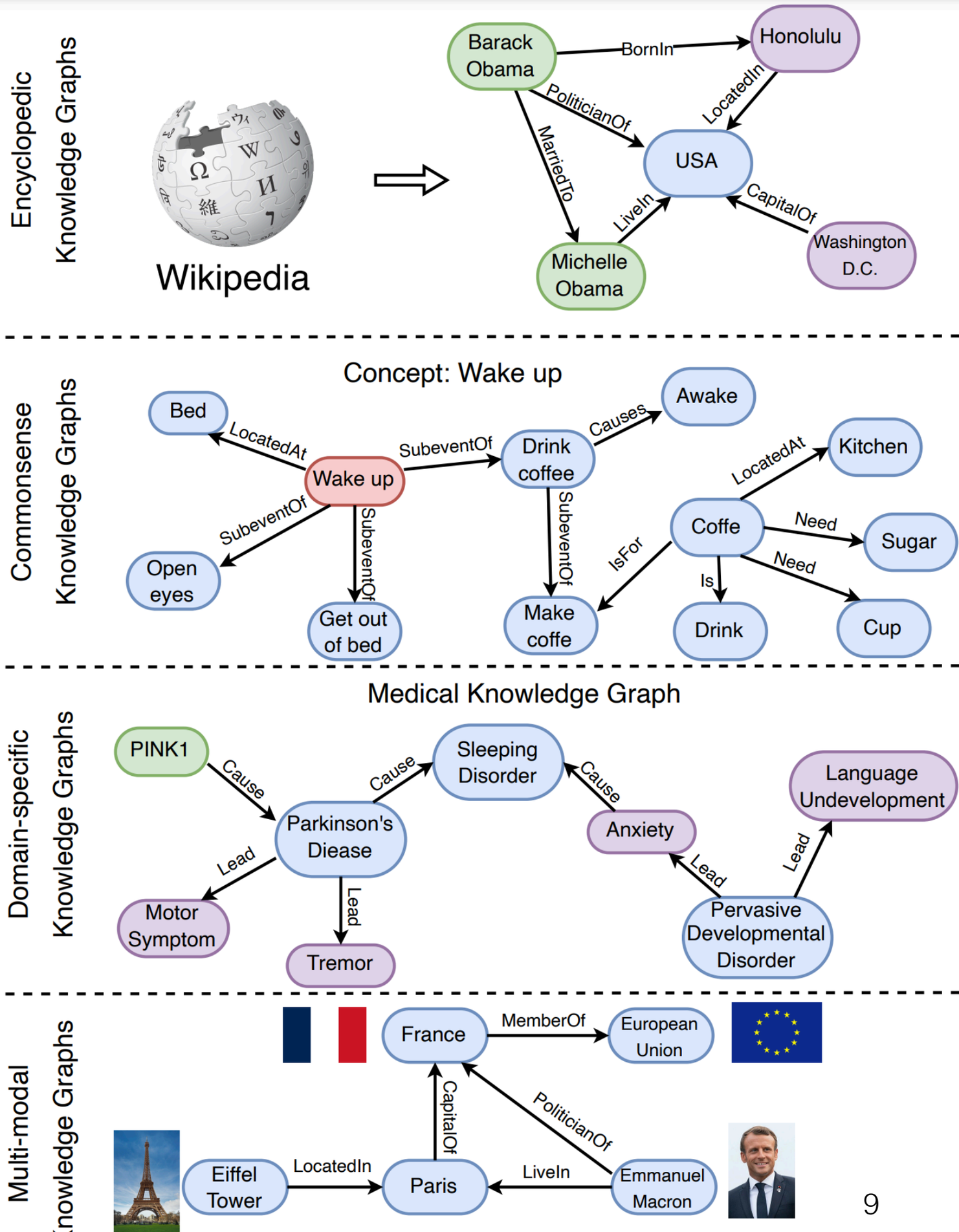
LLMs + KGs

- Research questions:
  - How can we **learn to create/expand knowledge graphs** with large neural networks?
  - How can we **learn from the information in knowledge graphs** to improve LLMs?
  - How can we **use structured knowledge to answer questions**?

Types of Knowledge Graphs

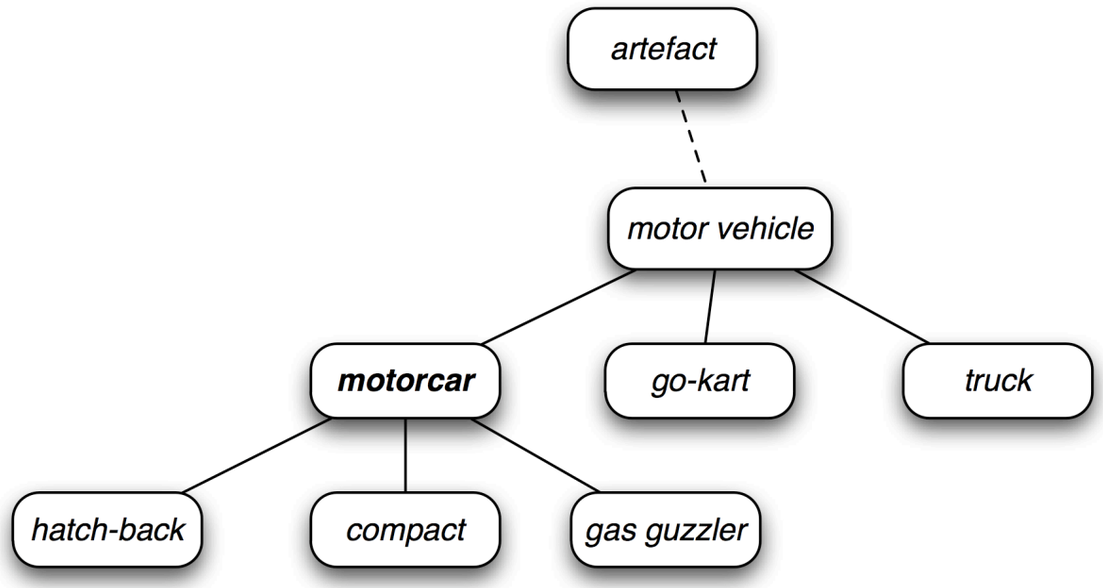
Four Major Types of KGs

- **Encyclopedic KGs**
  - Extracted from *diverse sources* (human experts, databases, encyclopedias)
  - Examples: WikiData, Freebase, DBpedia, YAGO, NELL
- **Commonsense KGs**
  - *Daily concepts* (e.g., objects and events) and *tacit knowledge*.
  - Examples: ConceptNet, ATOMIC, ASER, TransOMCS, CausalBank
- **Domain-specific KGs**
  - Smaller but more accurate/reliable in a specific domain: medical, biology, and finance.
  - Example: UMLS (medical)
- **Multimodal KGs**
  - Multimodal facts (images, sounds, videos)
  - Examples: IMGpedia, MMKG,



WordNet (Miller 1995)

- WordNet is a large database of words including parts of speech, semantic relations



- Nouns: is-a relation (hatch-back/car), part-of (wheel/car), type/instance distinction
- Verb relations: ordered by specificity (communicate -> talk -> whisper)
- Adjective relations: antonymy (wet/dry)

Image Credit: NLTK

Cyc (Lenant 1995)

- A manually curated database attempting to encode all common sense knowledge, 30 years in the making

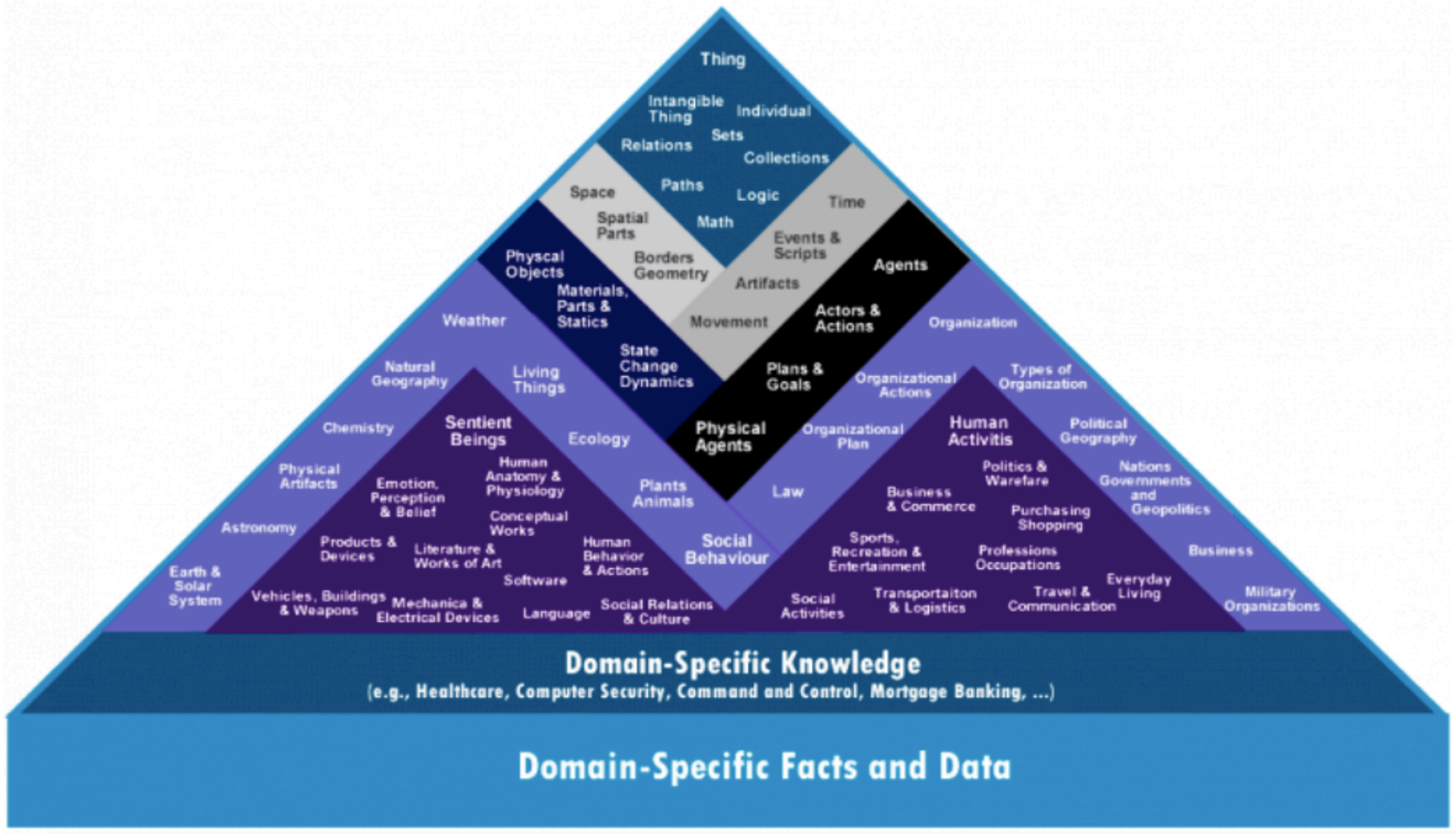


Image Credit: NLTK

DBPedia (Auer et al. 2007)

- Extraction of structured data from Wikipedia

Structured data

Carnegie Mellon University

From Wikipedia, the free encyclopedia

Carnegie Mellon University (**Carnegie Mellon** or **CMU** /ˈkɑːrnɪgi ˈmɛlən/ or /kɑːrˈneɪgi ˈmɛlən/) is a private research university in Pittsburgh, Pennsylvania.

Founded in 1900 by Andrew Carnegie as the Carnegie Technical Schools, the university became the Carnegie Institute of Technology in 1912 and began granting four-year degrees. In 1967, the Carnegie Institute of Technology merged with the Mellon Institute of Industrial Research to form Carnegie Mellon University.

The university's 140-acre (57 ha) main campus is 3 miles (5 km) from Downtown Pittsburgh. Carnegie Mellon has seven colleges and independent schools: the College of Engineering, College of Fine Arts, Dietrich College of Humanities and Social Sciences, Mellon College of Science, Tepper School of Business, H. John Heinz III College of Information Systems and Public Policy, and the School of Computer Science. The university also has campuses in Qatar and Silicon Valley, with degree-granting programs in six continents.

Carnegie Mellon is ranked 25th in the United States and 77th in the world by *U.S. News & World Report*.[9] It is home to the world's first degree-granting Robotics and Drama programs,[10] as well as one of the first Computer Science departments.[11] The university was ranked 89th for R&D in 2015 having spent $242 million.[12]

Carnegie Mellon counts 13,650 students from 114 countries, over 100,000 living alumni, and over 5,000 faculty and staff. Past and present faculty and alumni include 20 Nobel Prize Laureates,[13] 12 Turing Award winners, 22 Members of the American Academy of Arts & Sciences,[14] 19 Fellows of the American Association for the Advancement of Science, 72 Members of the National Academies, 114 Emmy Award winners, 44 Tony Award laureates, and 7 Academy Award winners.[15]

Coordinates: 40.443322°N 79.943583°W

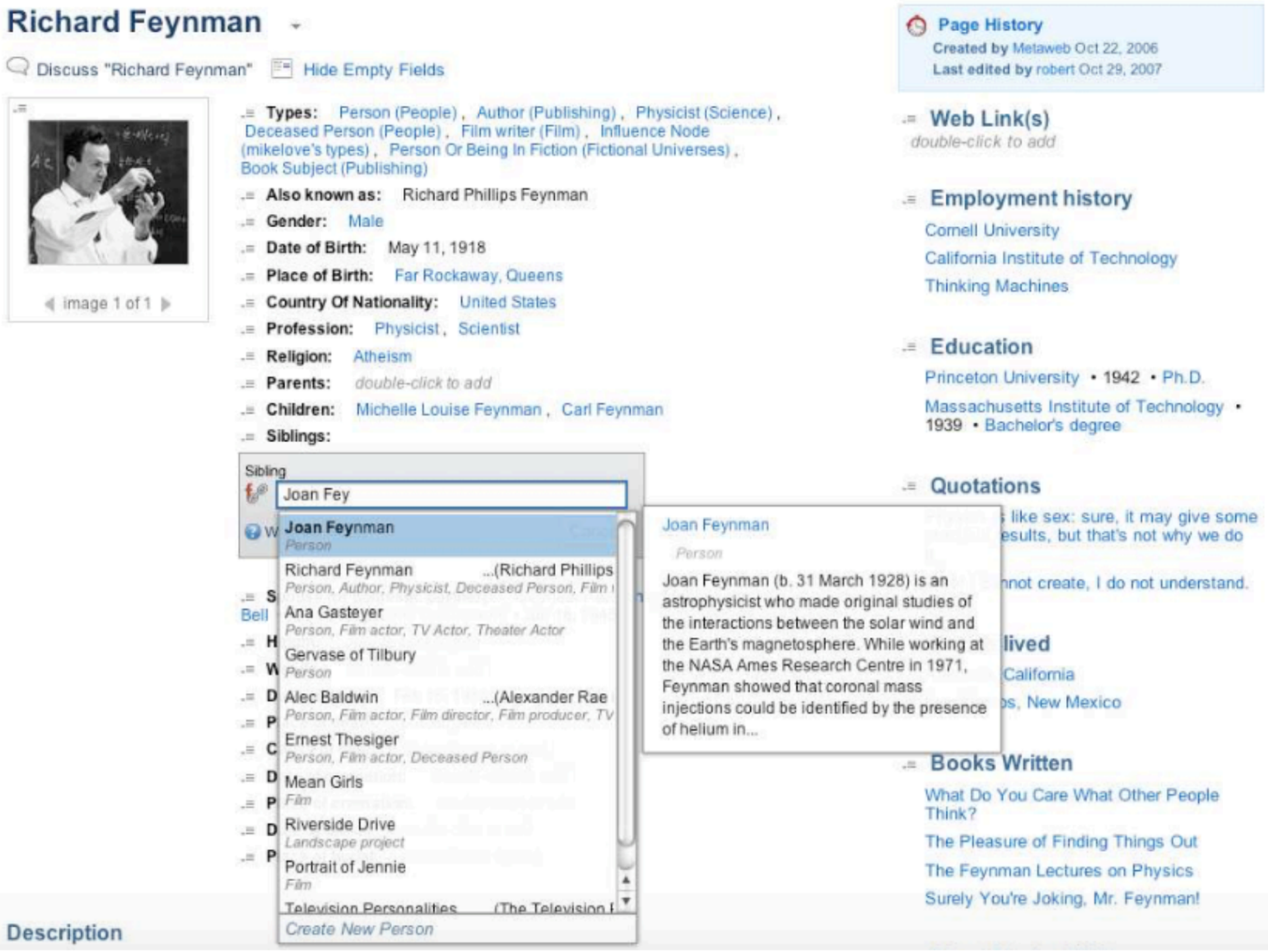
Carnegie Mellon University

Former names	Carnegie Technical Schools (1900–1912) Carnegie Institute of Technology (1912–1967) Carnegie-Mellon University (1968–1988) [1] Carnegie Mellon University (1988–present)
Motto	"My heart is in the work" (Andrew Carnegie)
Type	Private university
Established	1900 by Andrew Carnegie

- owl:Thing
- dul:Agent
- dul:SocialPerson
- wikidata:Q24229398
- wikidata:Q3918
- wikidata:Q43229
- dbo:Agent
- dbo:EducationalInstitution
- dbo:Organisation
- dbo:University
- geo:SpatialThing
- schema:CollegeOrUniversity
- schema:EducationalOrganization
- schema:Organization
- umbel-rc:Business
- umbel-rc:EducationalOrganization
- umbel-rc:Organization
- umbel-rc:University

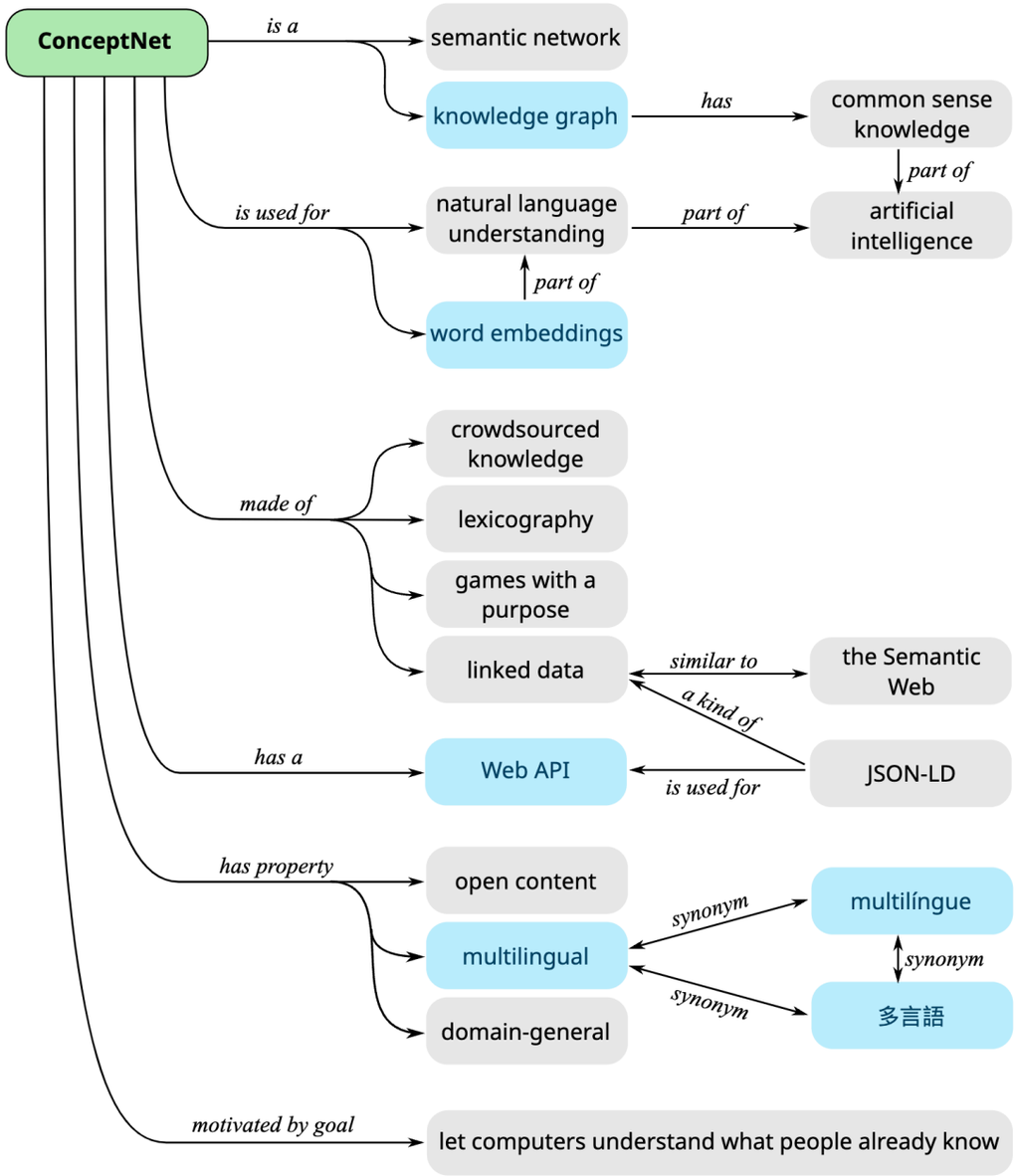
WikiData (Bollacker et al. 2008)

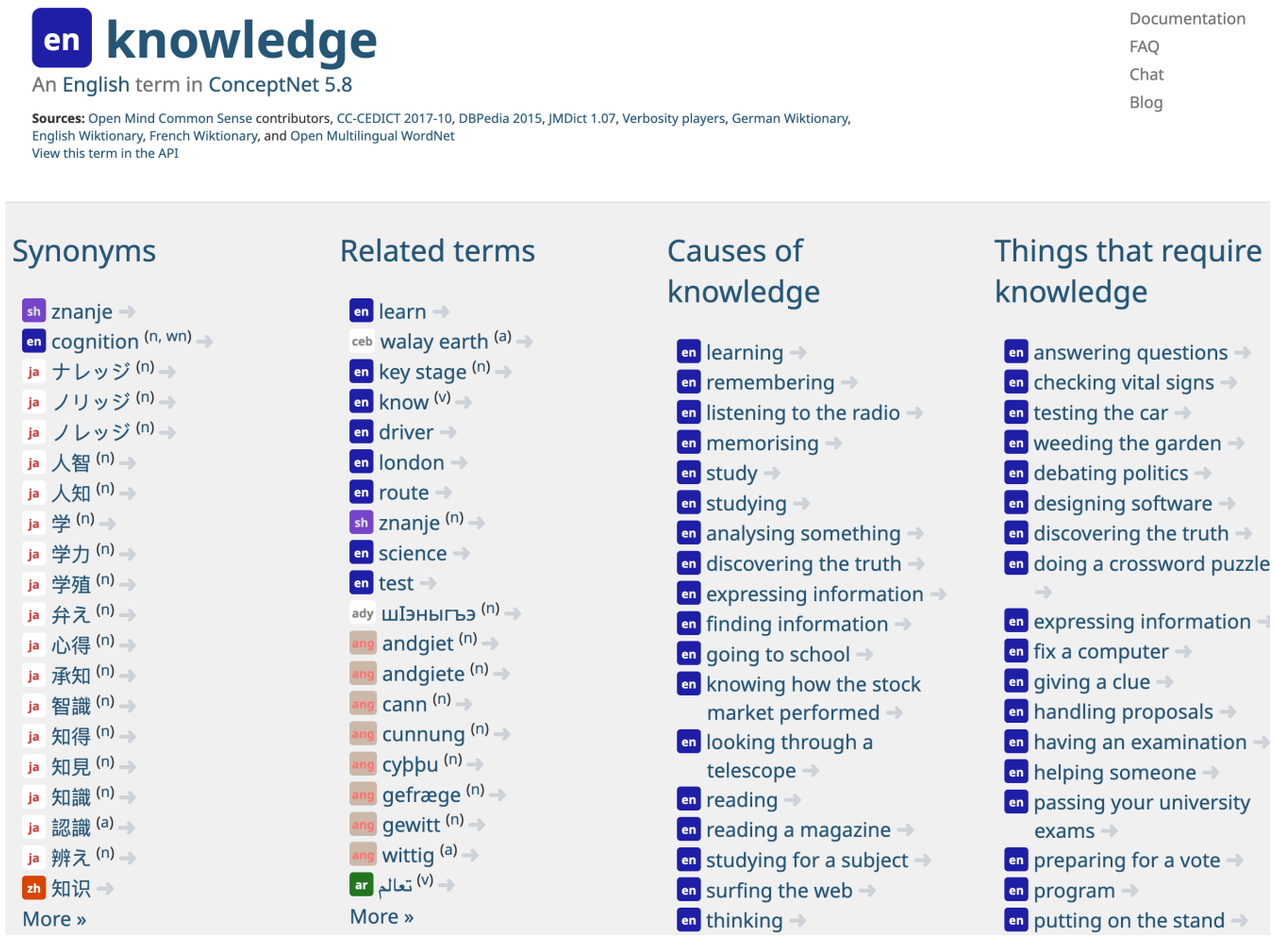
- *Curated* database of entities, linked, and extremely large scale, multilingual



ConceptNet

- An open, multilingual KG originated at MIT Media Lab in 1999
- It has since grown to include knowledge from other crowdsourced resources, expert-created resources, and games with a purpose.



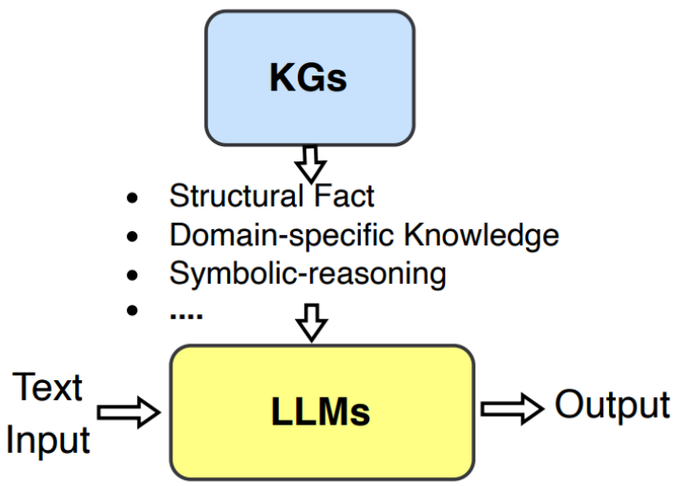


https://conceptnet.io/c/en/knowledge

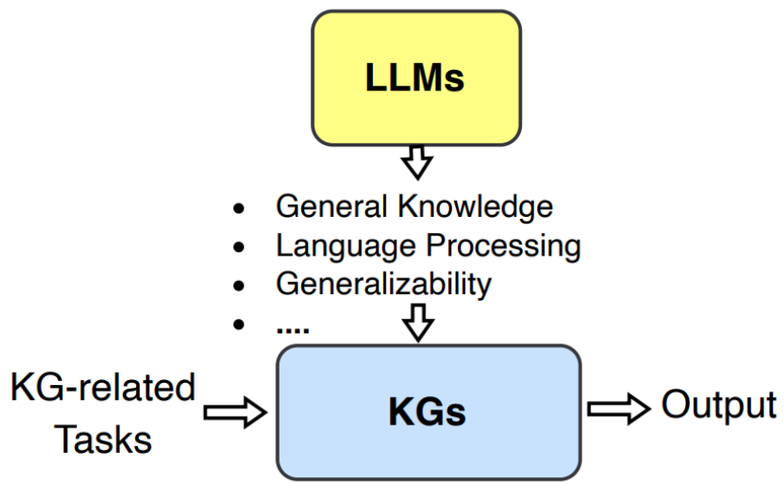
Unifying LLMs and KGs

General Roadmap of Unifying KGs and LLMs

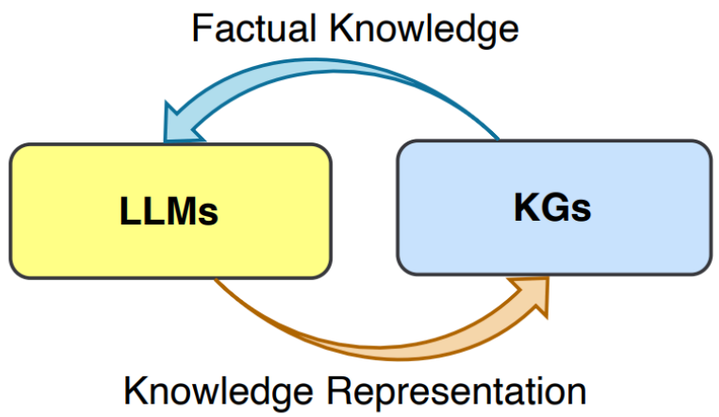
- Three general frameworks in the NLP/Data Mining communities
  - **KG-enhanced LLMs**: incorporate KGs into LLM pretraining/inference
  - **LLM-augmented KGs**: use LLMs to augment the incomplete KGs (improve KG representations, KG construction)
  - **Synergized LLMs + KGs**: mutually improve each other from all perspectives (including data, model, optimization and applications)



a. KG-enhanced LLMs

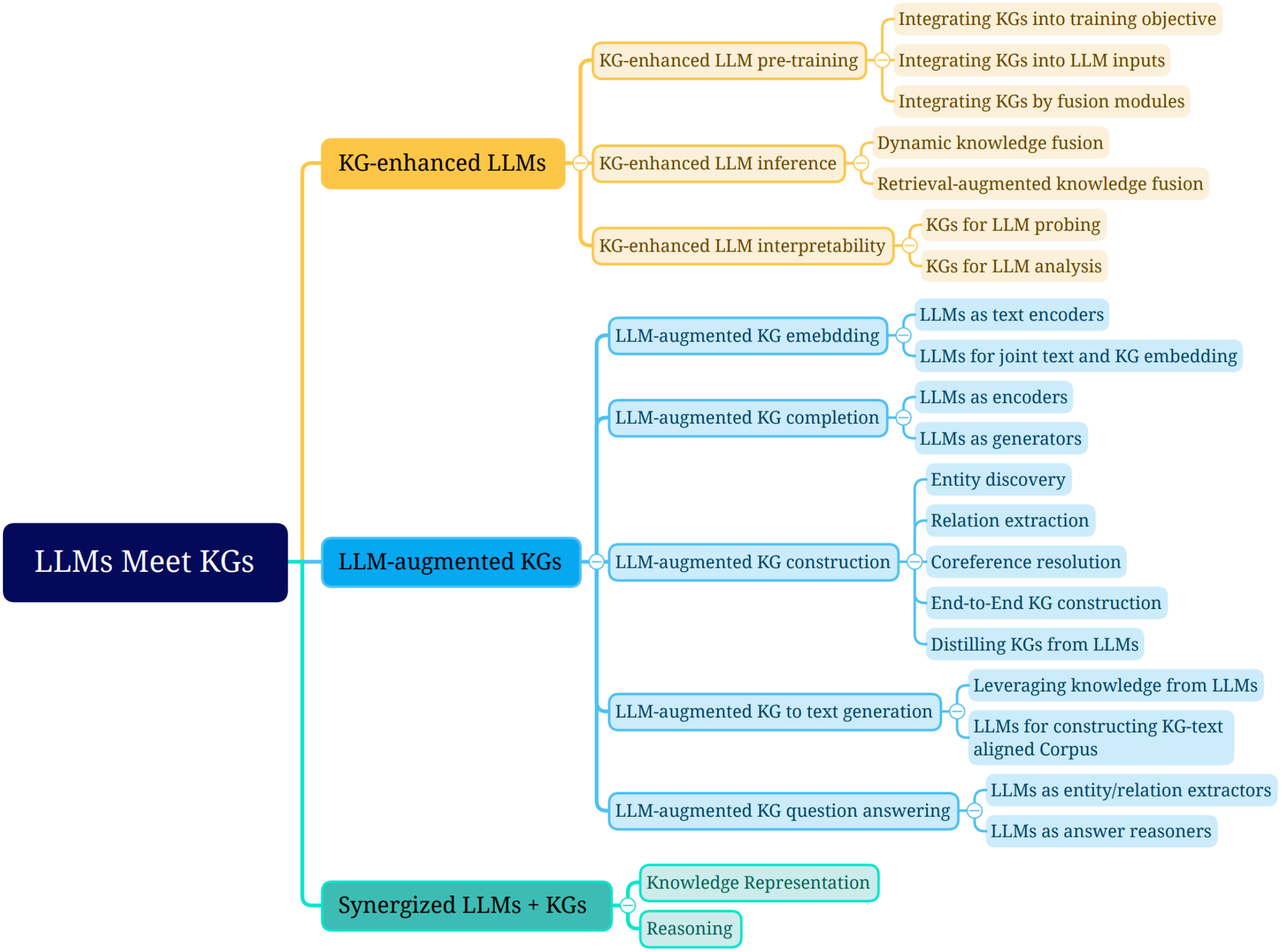


b. LLM-augmented KGs

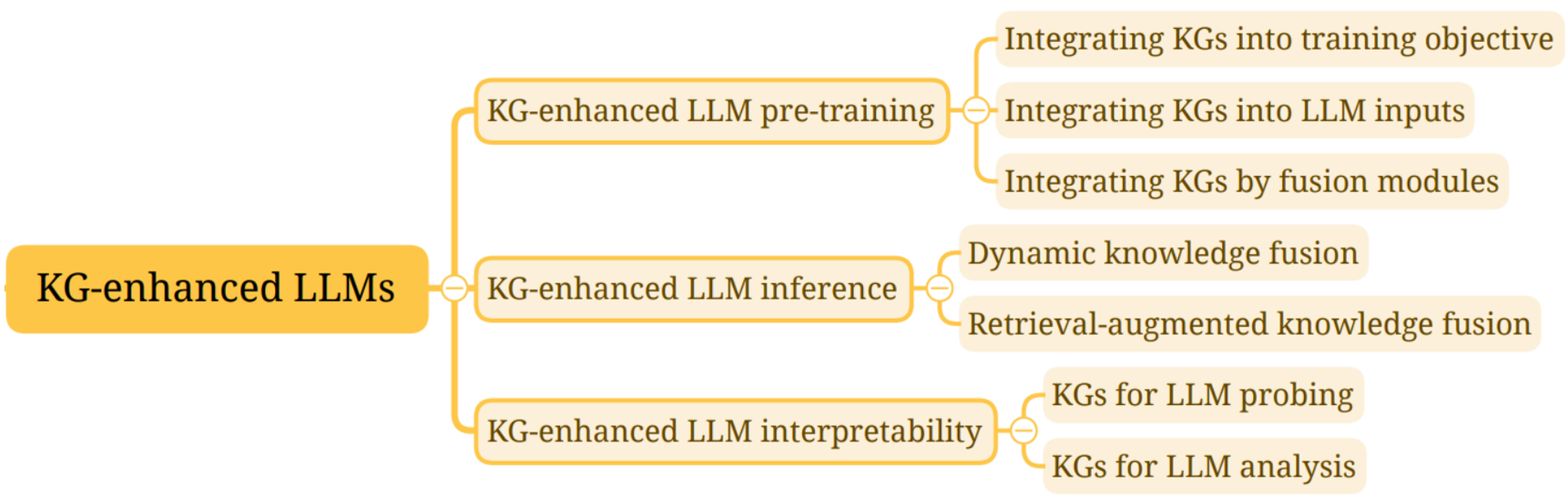


c. Synergized LLMs + KGs

Fine-grained Categorization of Existing and Ongoing Research



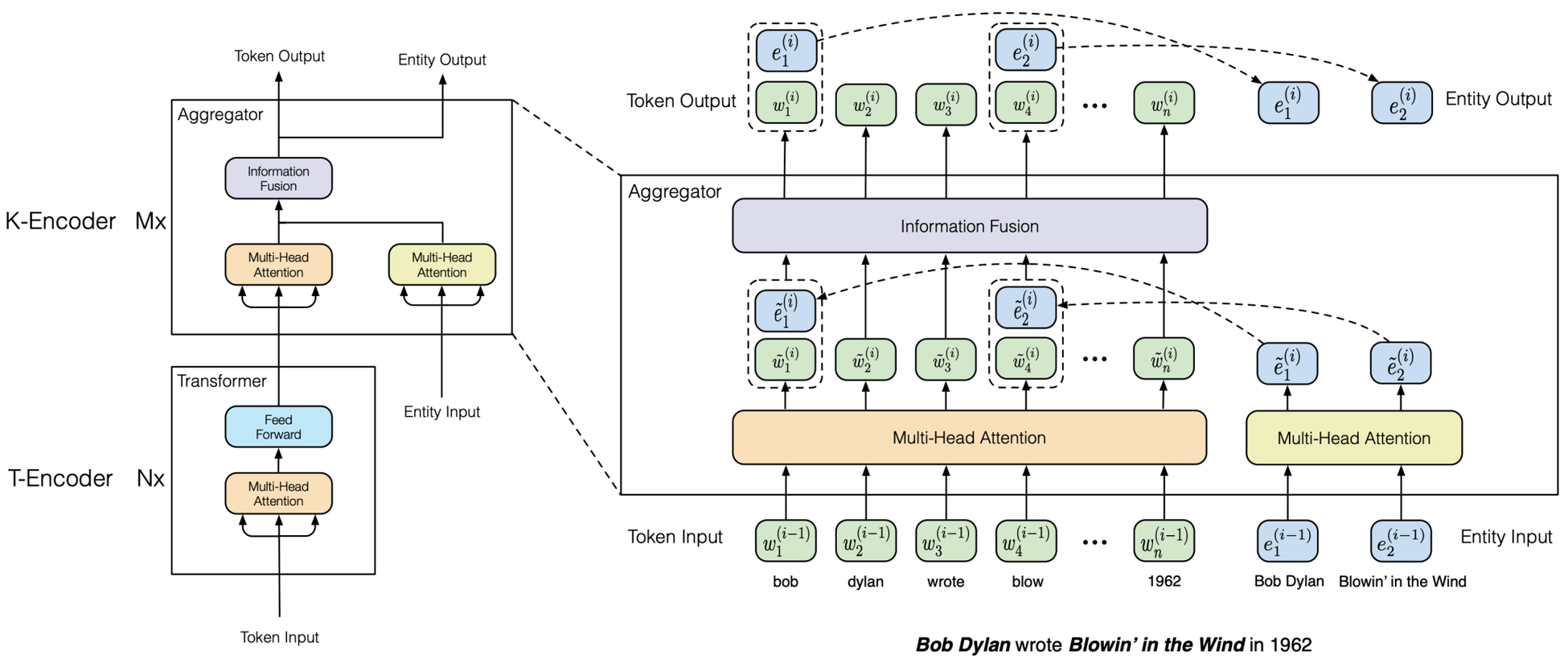
KG-enhanced LLMs



KG-enhanced LLM Pretraining

- **ERNIE: Denoising auto-encoder**
  - 5% of time, replace the entity with a random entity, and train the model to correct the errors
  - 15% of time, mask the token-entity alignment, train the model to predict the entity given the word
  - 80% of time, keep token-entity alignments unchanged, train the model to integrate entity info into token embedding for better language modeling.

$$p(e_j|w_i) = \frac{\exp(\texttt{linear}(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_j)}{\sum_{k=1}^{m} \exp(\texttt{linear}(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_k)},$$

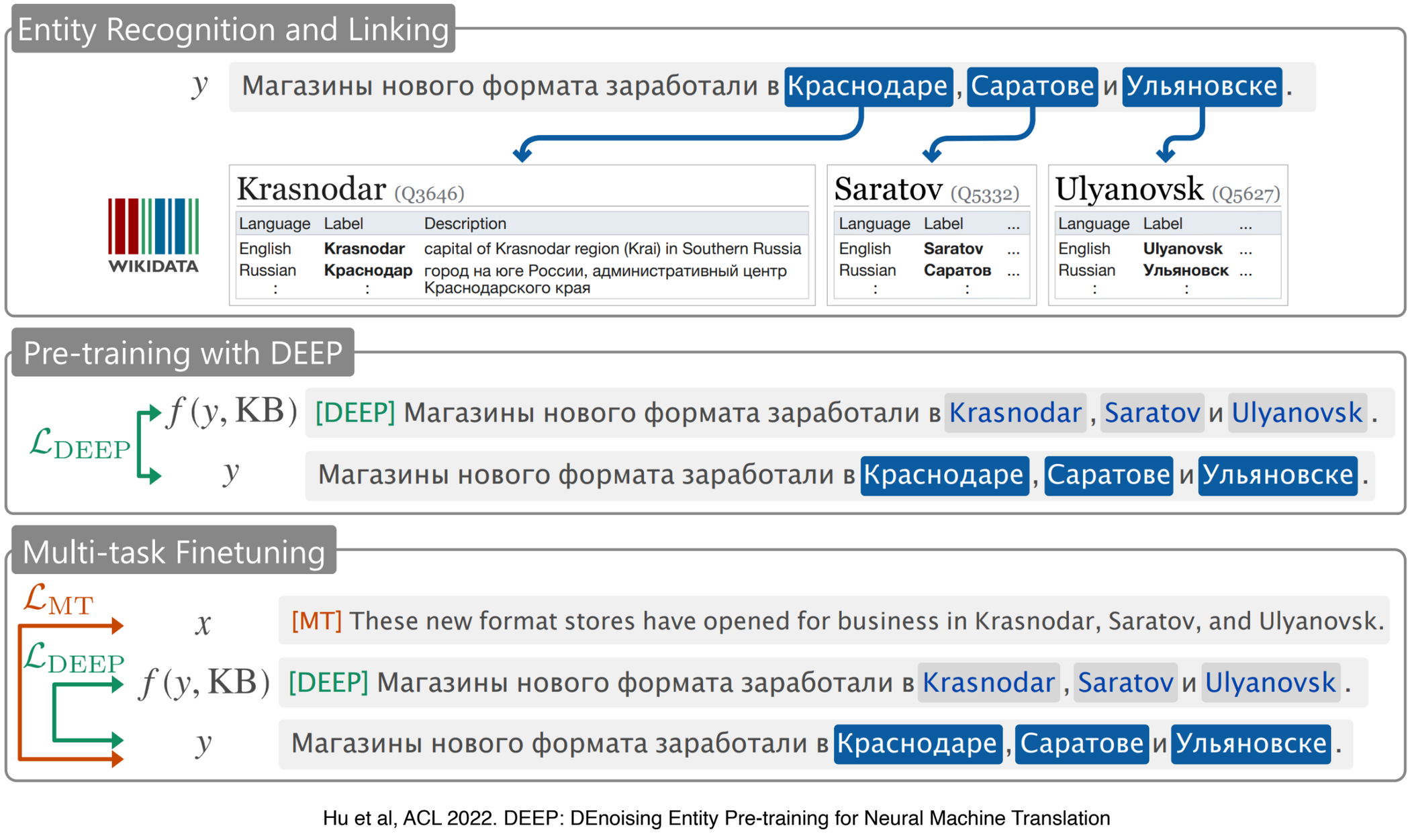


(a) Model Achitecture (b) Aggregator

Zhang et al. 2019. ERNIE: Enhanced Language Representation with Informative Entities

KG-enhanced LLM Pre-training

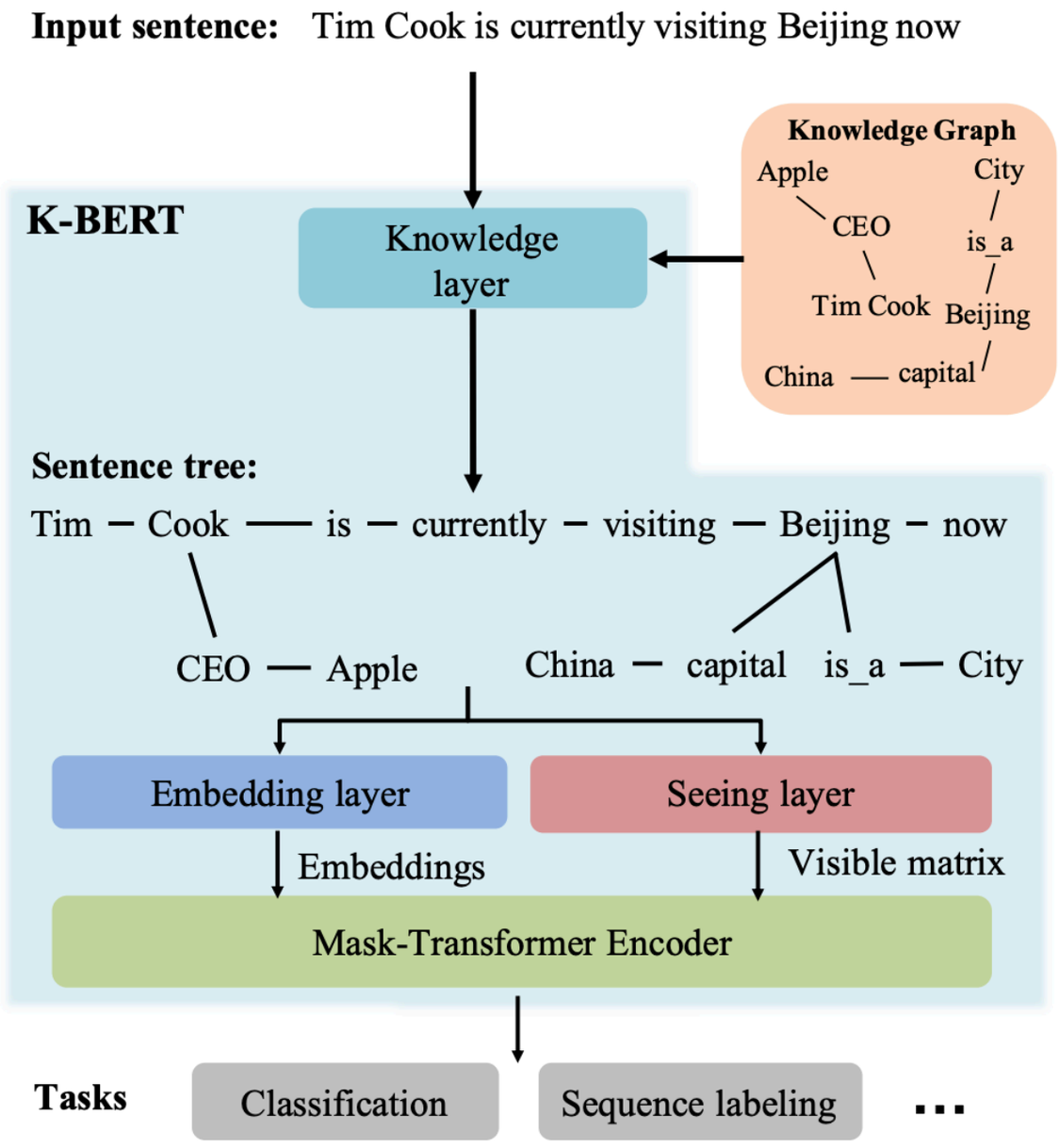
- **DEEP:** Integrate multilingual KG data to multilingual LLM pre-training:
  - Perform entity linking to detect mentions and link them to a KG
  - Obtain their multilingual translational information, add them to the pre-training data
  - Train an encoder-decoder model by machine translation and denoising auto-encoding objectives (multi-task learning)



Hu et al, ACL 2022. DEEP: DEnoising Entity Pre-training for Neural Machine Translation

K-BERT

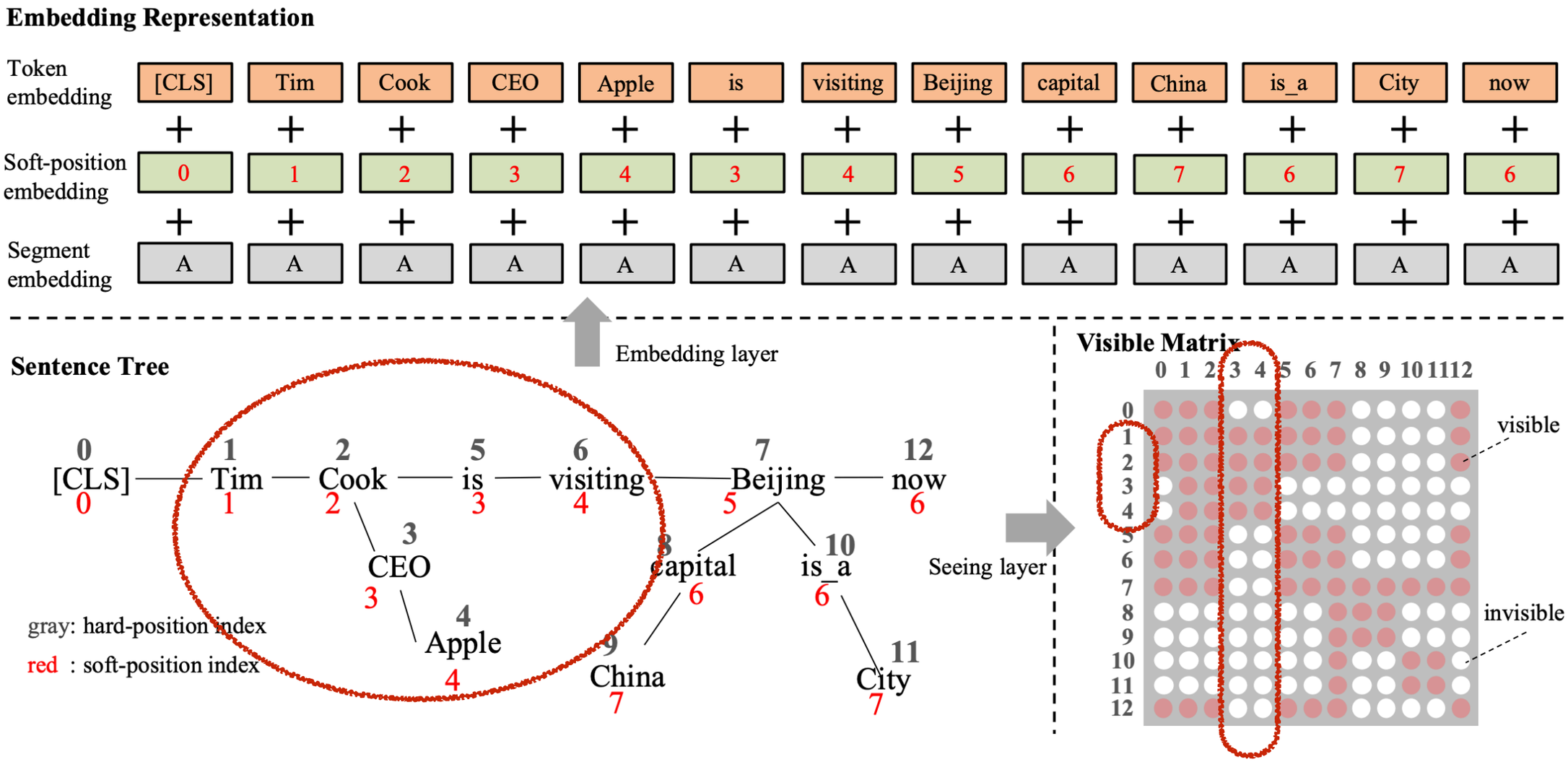
- Add some relation triplets to entity mentions in the sentence
  - (Tim Cook, CEO, Apple), (Beijing, capital, China), (Beijing, is_a, City)



Liu et al. 2019. K-BERT: Enabling Language Representation with Knowledge Graph

K-BERT

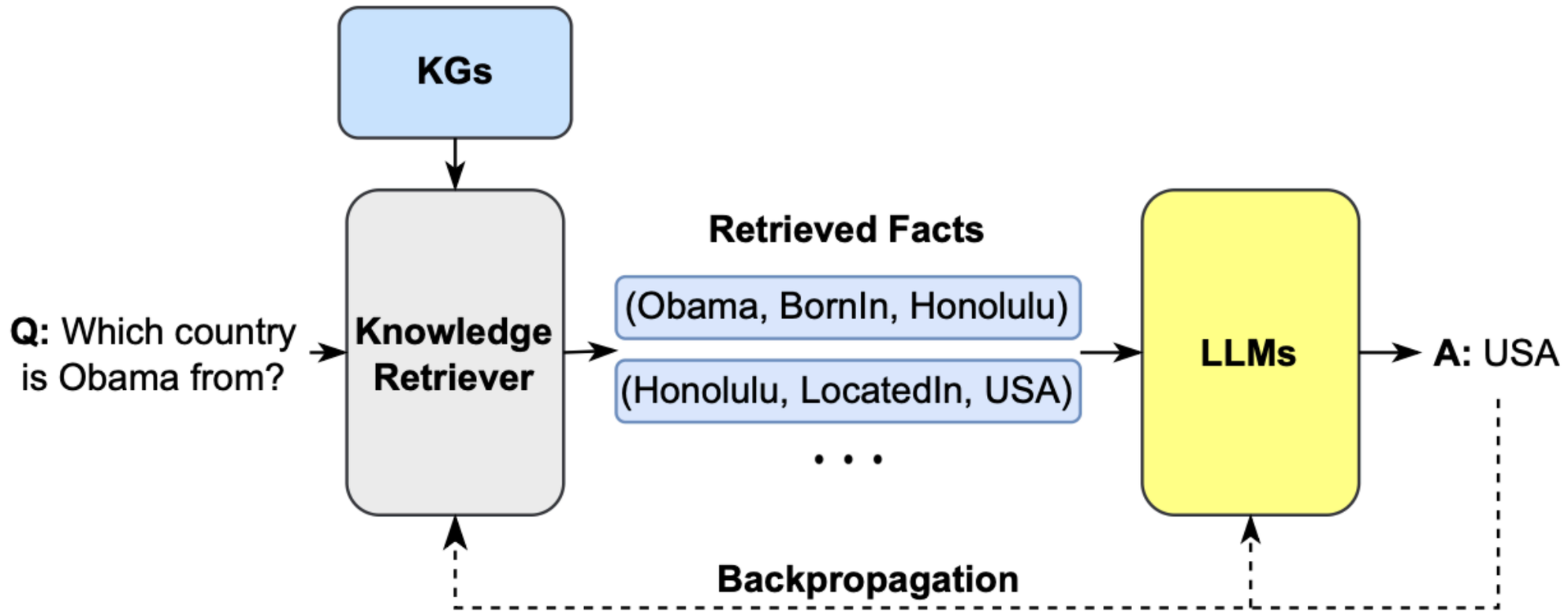
- Modify the position embedding of the added path
- Modify the attention masks so that only the entity mention (for example, **Tim Cook**) can attend to its corresponding relation triplet (**Tim Cook, CEO, Apple**).



Liu et al. 2019. K-BERT: Enabling Language Representation with Knowledge Graph

KG-enhanced LLM Inference

- At test time, retrieve related KG information to augment the LLM for prediction
- Most methods focus on OpenQA tasks, as the model requires up-to-date knowledge

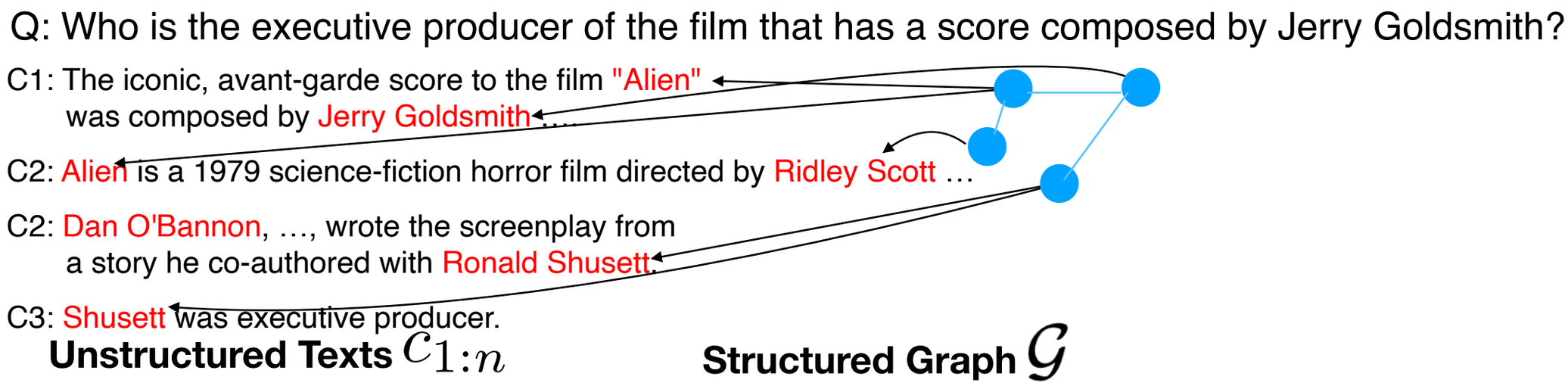


(Dotted line: Backpropagation is optional)

SeqGraph: Integrate KGs to prompt LLMs

- Given a text query q, we have access to a search engine to retrieve a set of related text passages $c_{1:n} = \{c_1, \cdots, c_n\}$ and an entity graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ constructed from the retrieved unstructured texts. We aim to use a LLM to estimate:

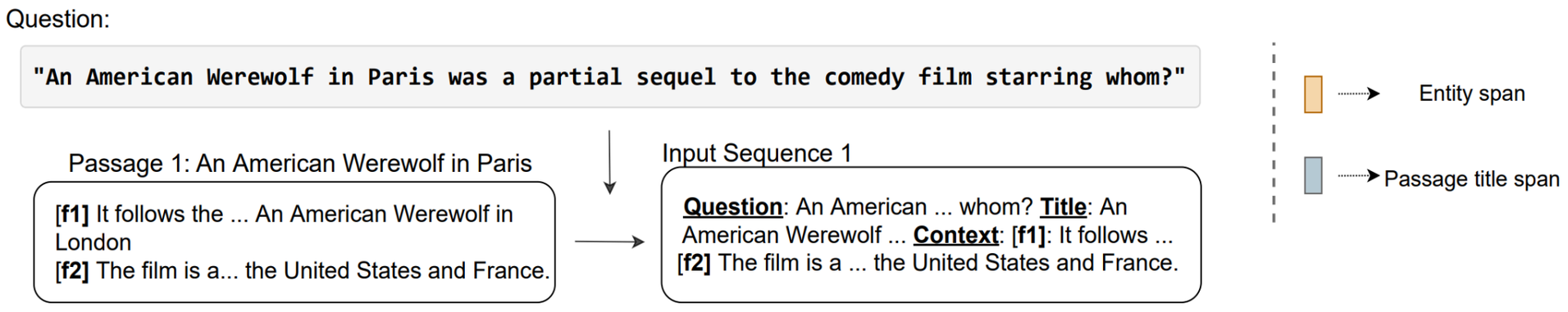
Reasoning probability $p(r|q, c_{1:n}, \mathcal{G})$ *Answer probability* $p(a|q, c_{1:n}, \mathcal{G}, r)$



[Ramesh, Sreedhar, **Hu**. ACL 2023] Single Sequence Prediction over Reasoning Graphs for Multi-hop QA

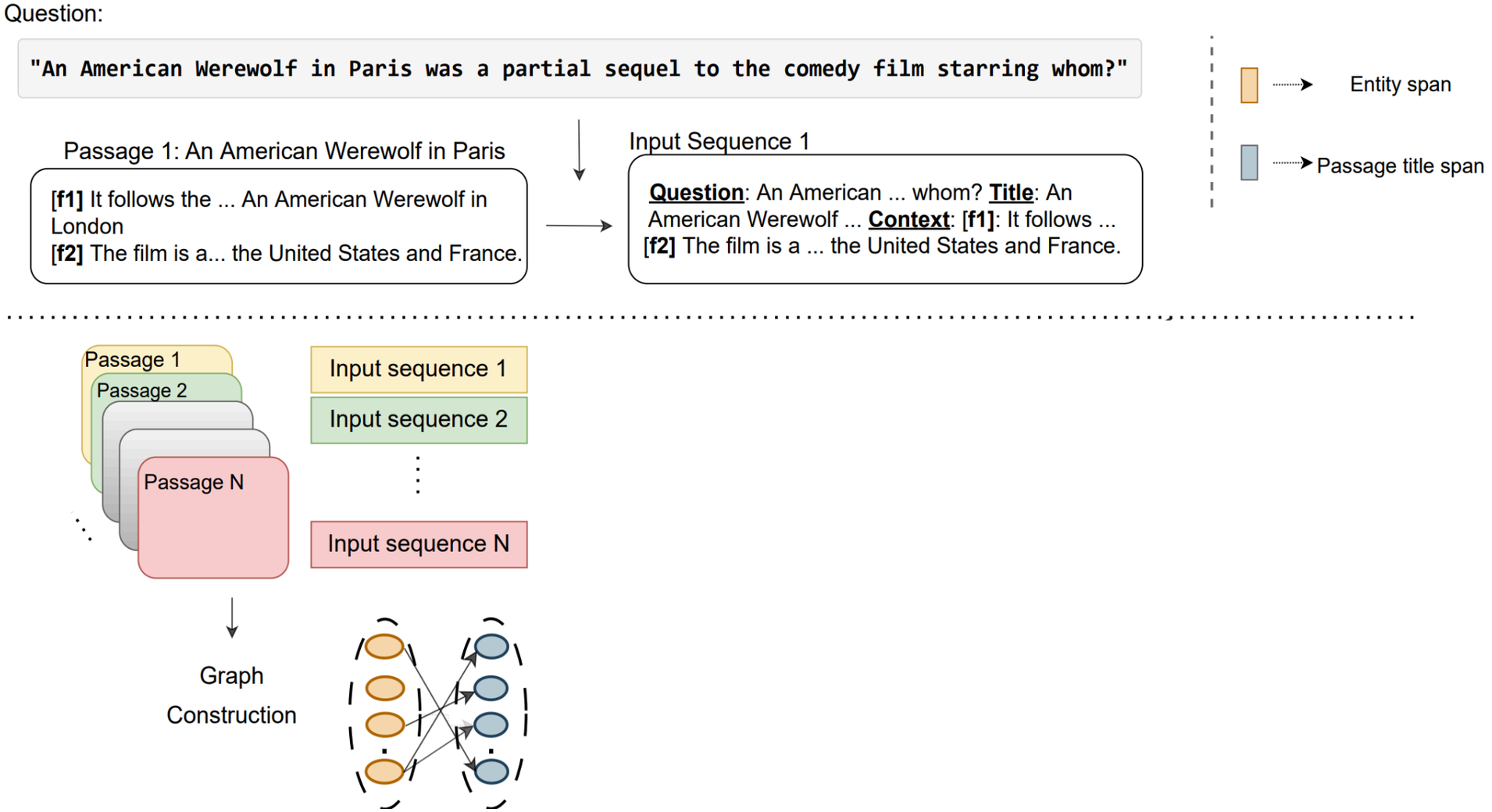
SeqGraph Workflow

- Combine retrieved context passages with the question as input to LLMs



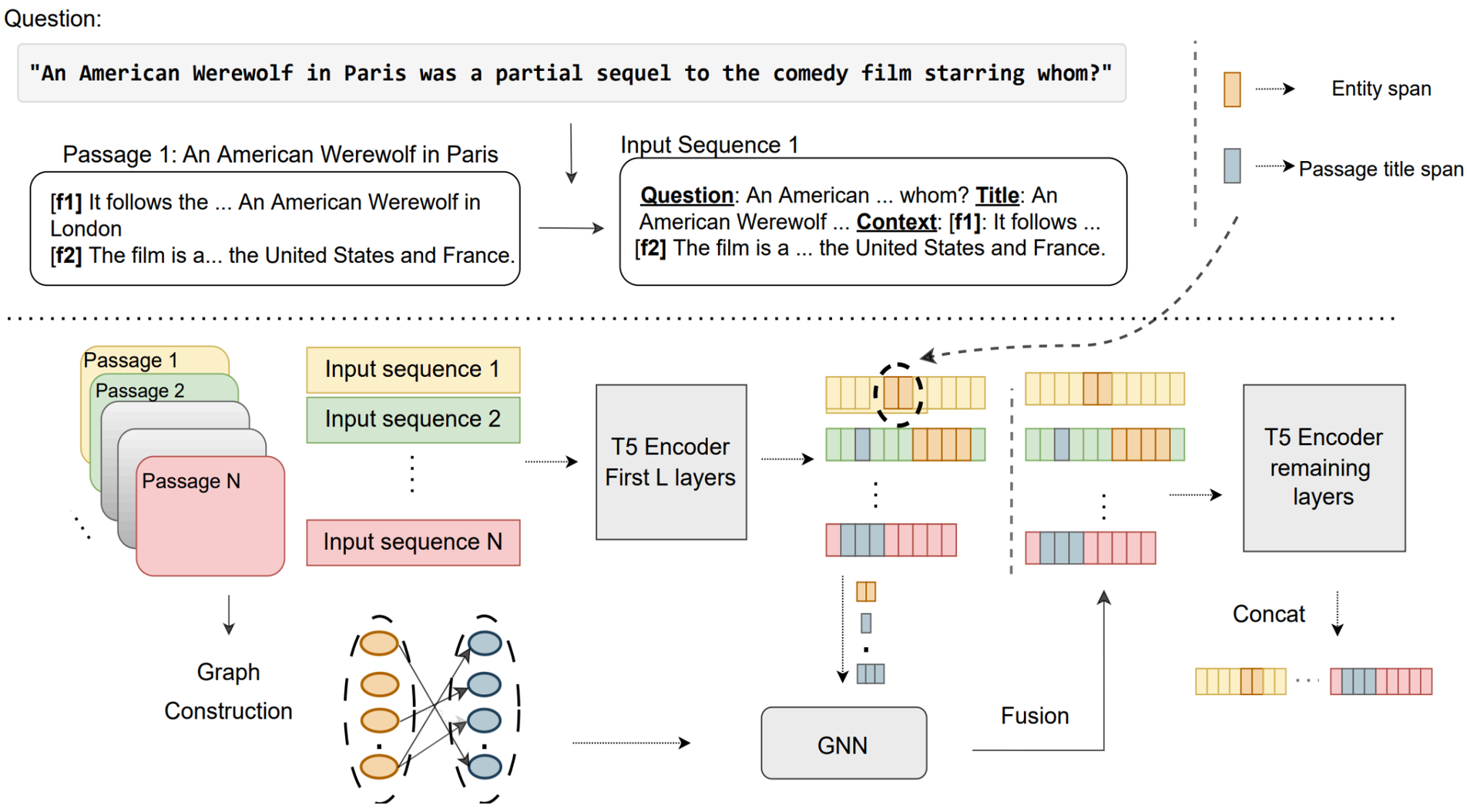
SeqGraph Workflow

- Identify entity mentions from the retrieved context passages, link them to WikiData, and construct a local knowledge graph



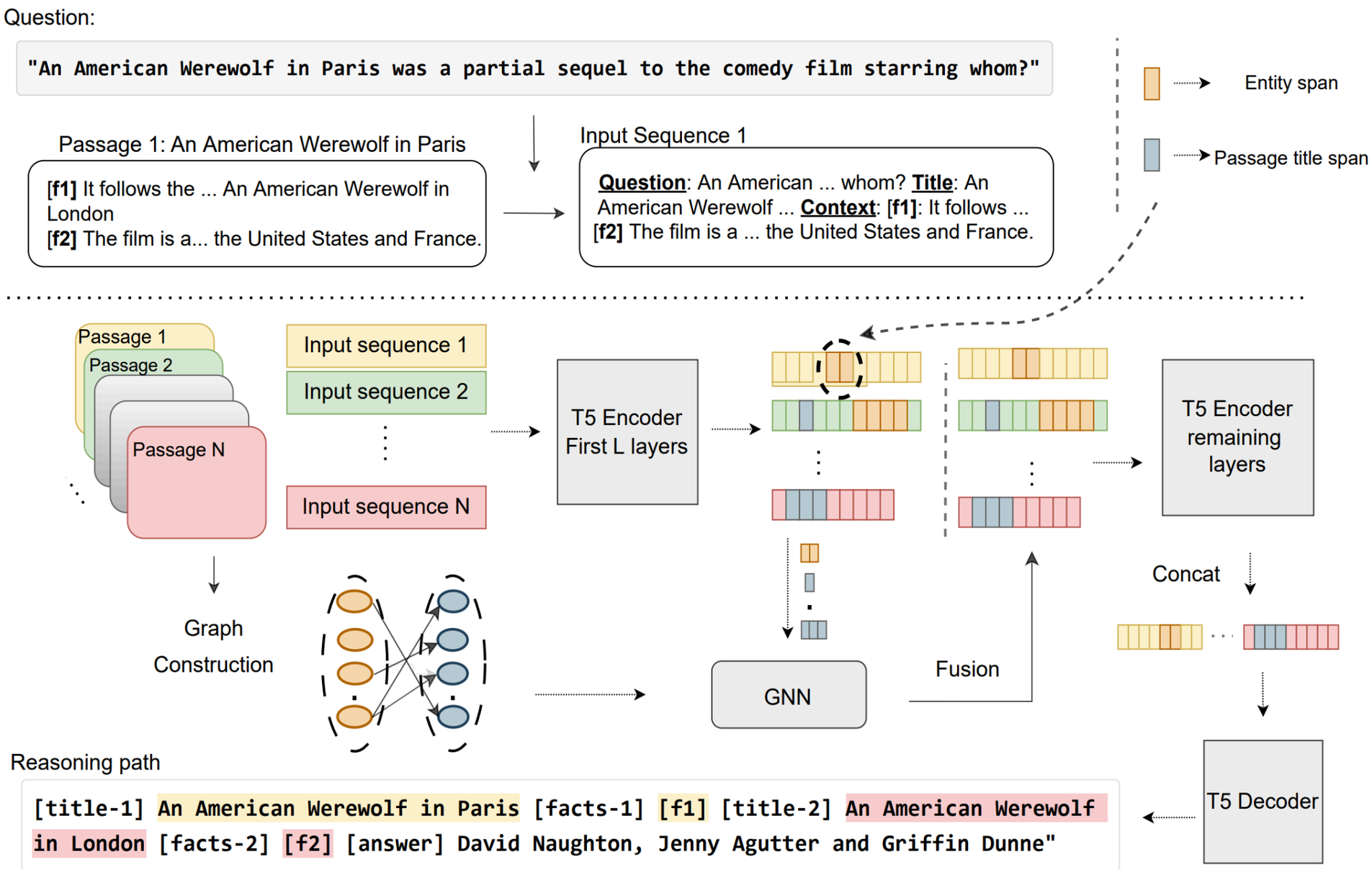
SeqGraph Workflow

- Use a LLM to encode text, and a GNN to encode KG, and fuse their representations



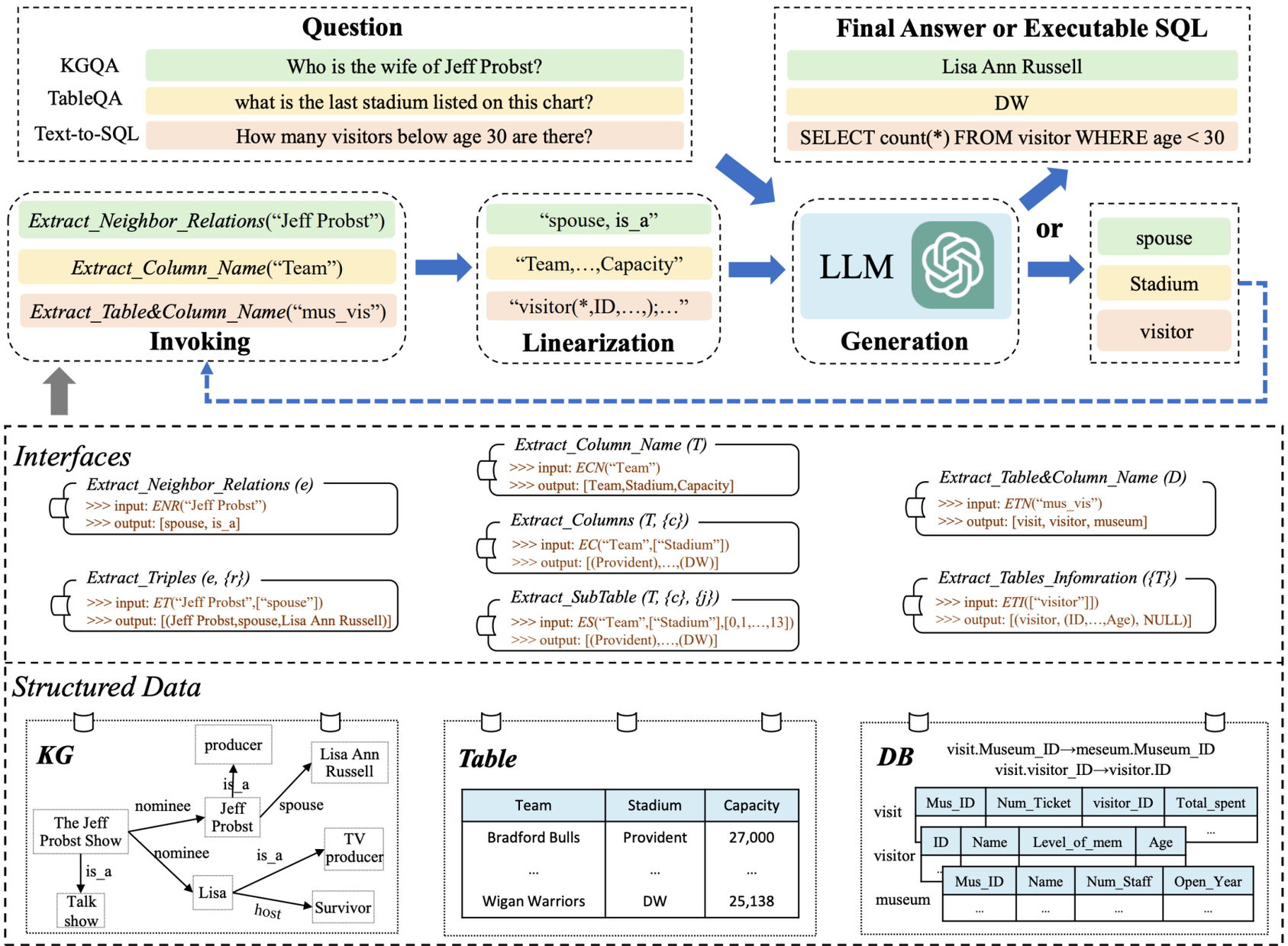
SeqGraph Workflow

- Feed the fused representations to a LLM decoder to generate both reasoning path, and the final answer to a multi-hop question



StructGPT

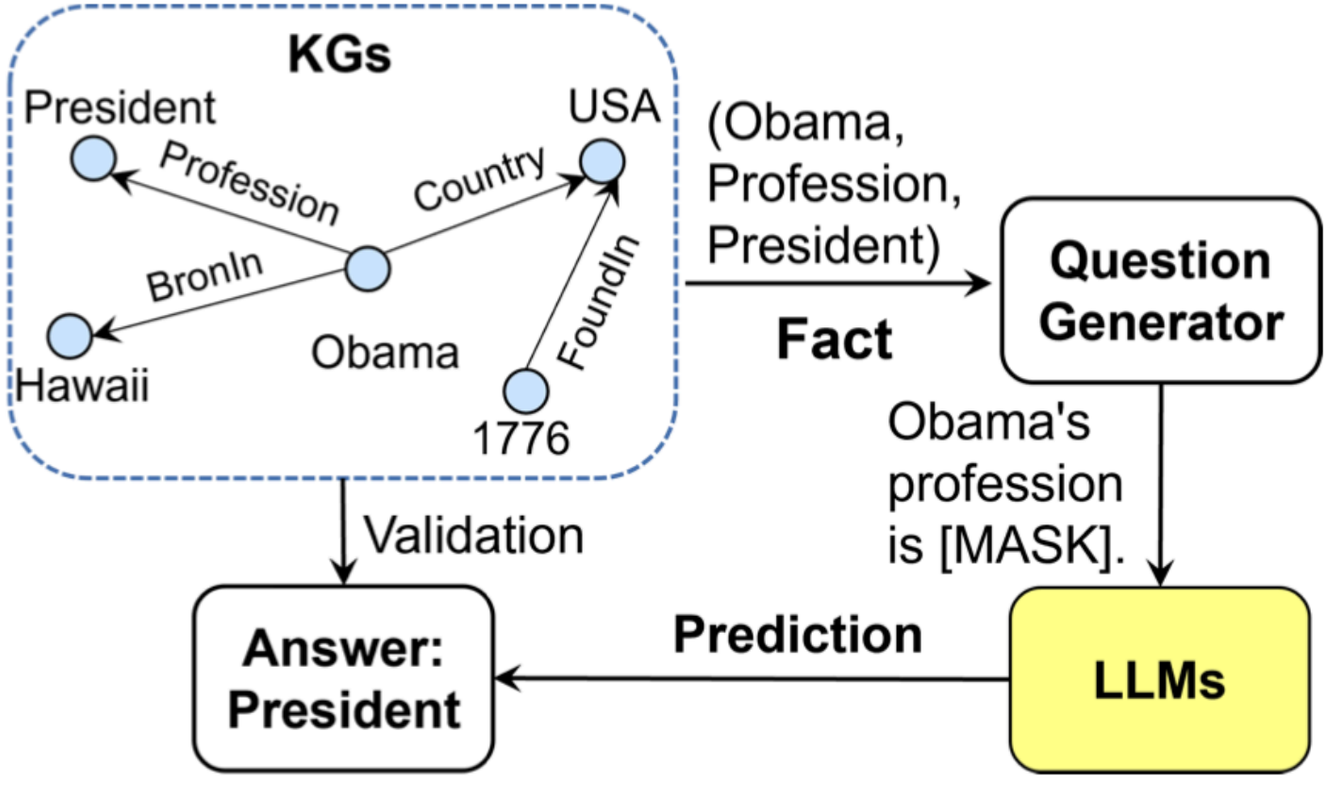
- **Prompt engineering without fine-tuning**
  - Invoking: retrieve related information from structured data
  - Linearization: Convert retrieved information into a text sequence
  - Generate: ask LLMs to generate either an answer or an executable SQL comment



Jiang et al. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data

KG-enhanced LLM Interpretability

- **KGs for LLM Probing**: Use KGs to understand LLMs' embedded knowledge. LAMA is the first study, and follow-up works include LPAQA, BioLAMA, MedLAMA
  - Sample some relation triplets, and construct some simple questions with a [MASK] token
  - Ask the LLMs to predict the [MASK] token
  - Use the ground-truth relation triplets to validate the accuracy

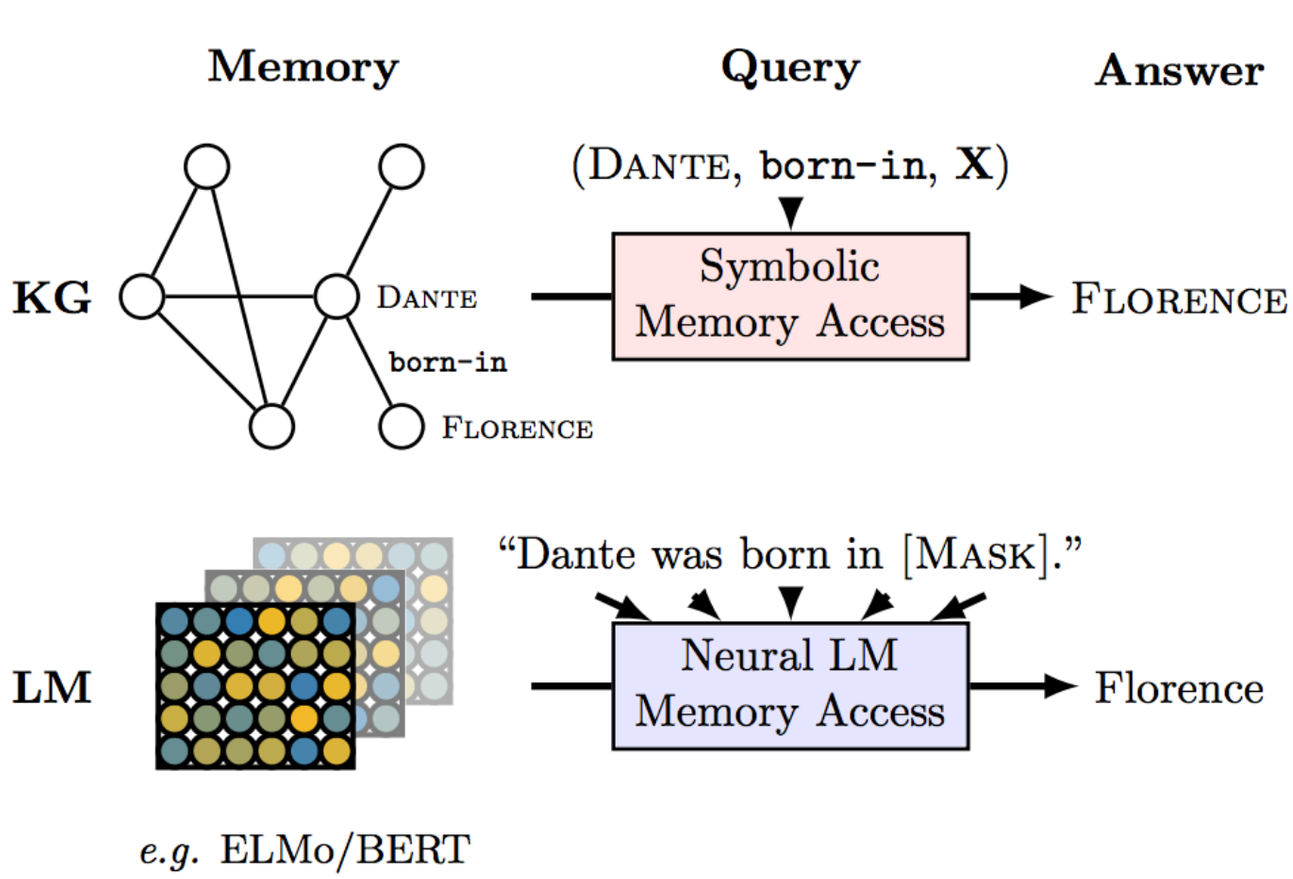


Petroni et al. 2019. Language Models as Knowledge Bases?

LAMA: LMs as KBs?

(Petroni et al. 2019)

- Structured queries (e.g., SQL) to query KBs.
- Natural language prompts to query LMs.



Petroni et al. 2019. Language Models as Knowledge Bases?

LMs as KBs?
(Petroni et al. 2019)

- LAMA benchmark
  - Manual prompts for 41 relations: “[X] was founded in [Y].”
  - Fill in subjects and have LMs (e.g., BERT) predict objects: “Bloomberg L.P. was founded in [MASK].”
- Accuracy: ELMo 7.1%, Transformer-XL 18.3%, BERT-base 31.1%

Mask 1 Predictions:

5.2% **Chicago**

4.1% **London**

2.8% **Toronto**

2.3% **c**

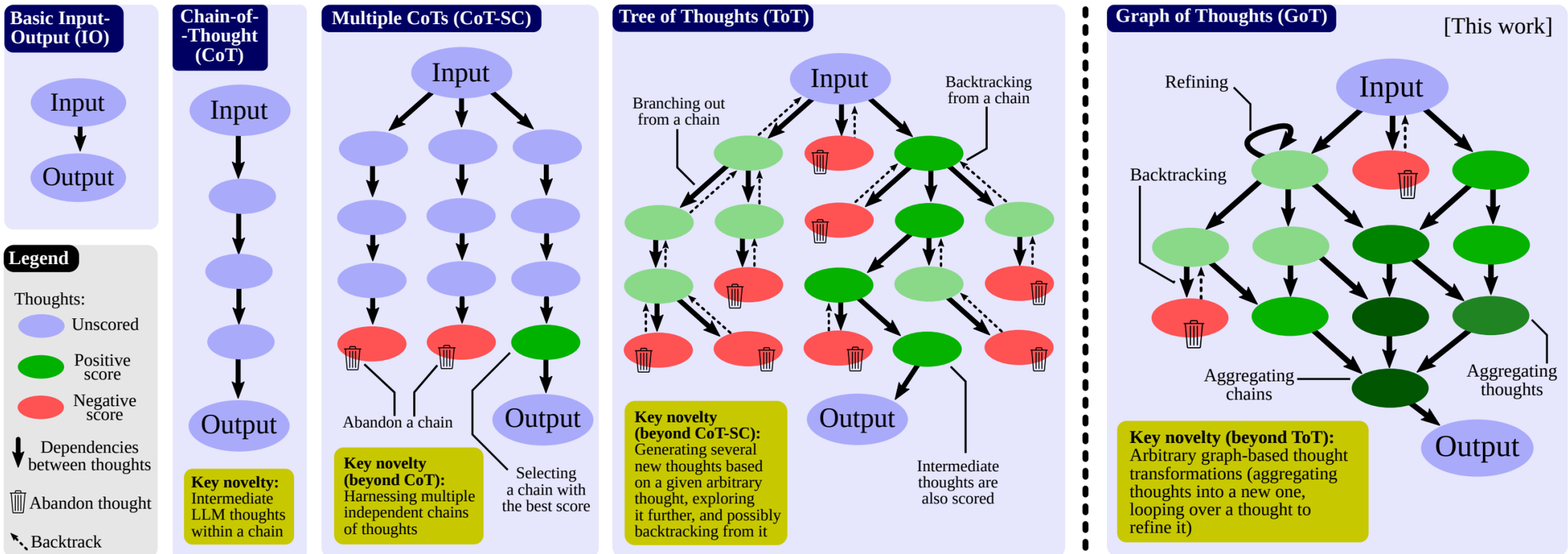
1.6% **India**

https://demo.allennlp.org/masked-lm/s/bloomberg-lp-was-founded-mask/I5Q1P2T5Z0

Structured Prompting Strategies

- In addition to prompting LLMs to generate an entity in a relation triplet (LAMA), recent structured prompting methods prompt the LLMs to generate a long sequence in a linear chain (Chain-of-thought), a tree (Tree-of-thought) or a graph (Graph-of-thought).

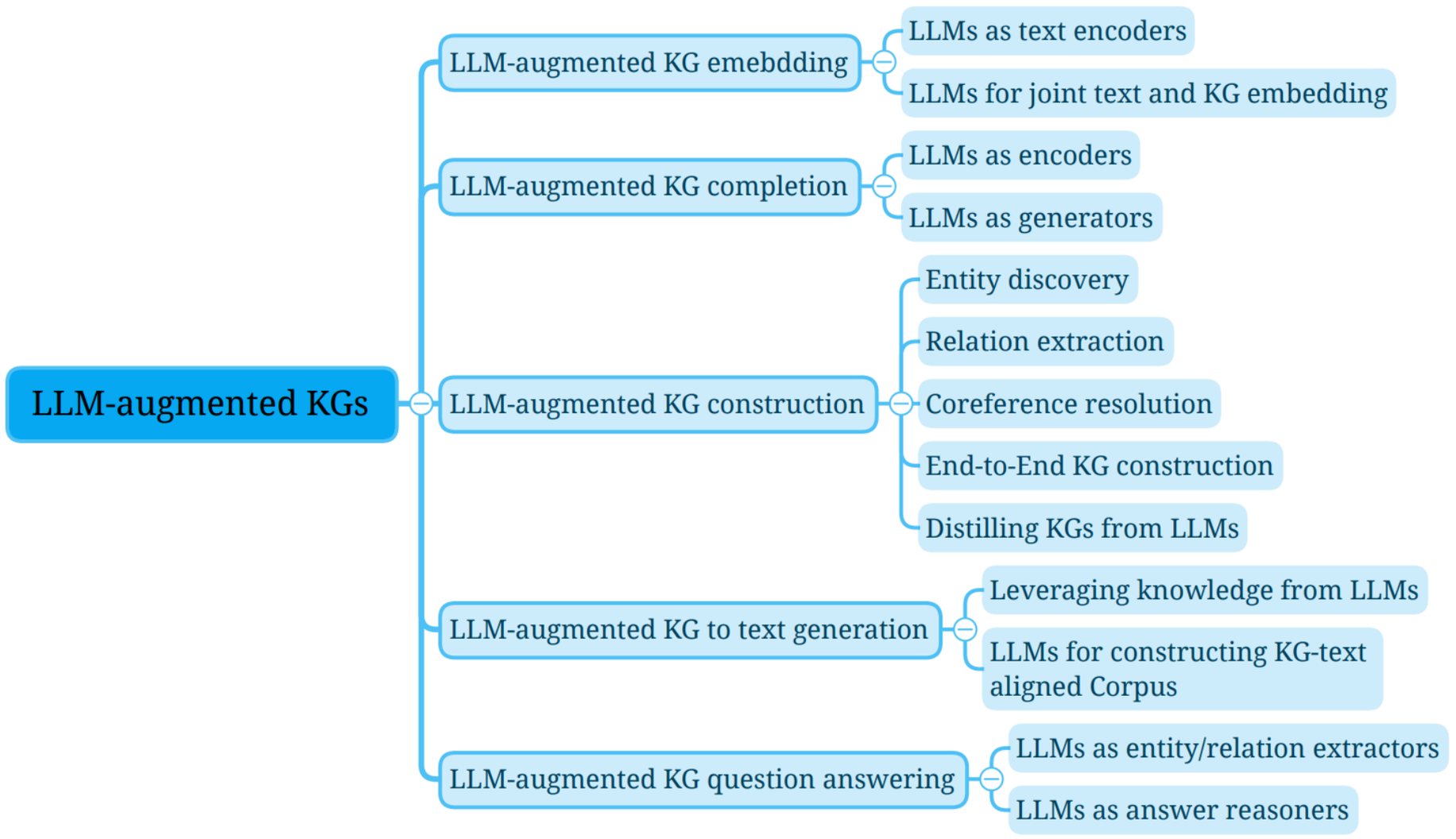
- Essentially, **probe the LLMs to recover some paths in a KG**



LLM-augmented KGs

LLM-augmented KGs

- Use LLM to improve KG tasks:

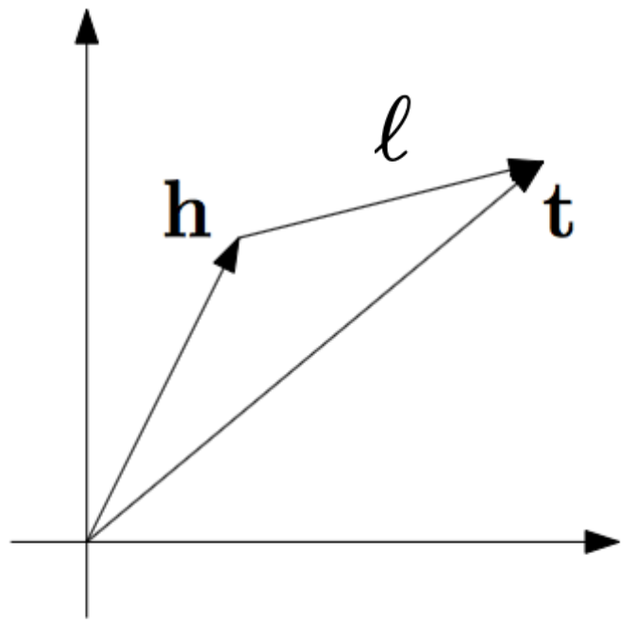


Task 1: Learning KG embeddings — before LLMs

- **Learning node/edge embeddings for a KG before LLMs**: rely more on **structural** information of KGs than **semantic** information to optimize a scoring function computed from embeddings
  - Examples: TransE, and DisMult
  - Limitations: unseen entities and long-tailed relations due to their limited structural connectivity

- **TransE**: express triples as additive transformation
- **Objective**: minimize the distance of existing triples with a margin-based loss that

$$\sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{\ell},\boldsymbol{t'})\right]_+$$



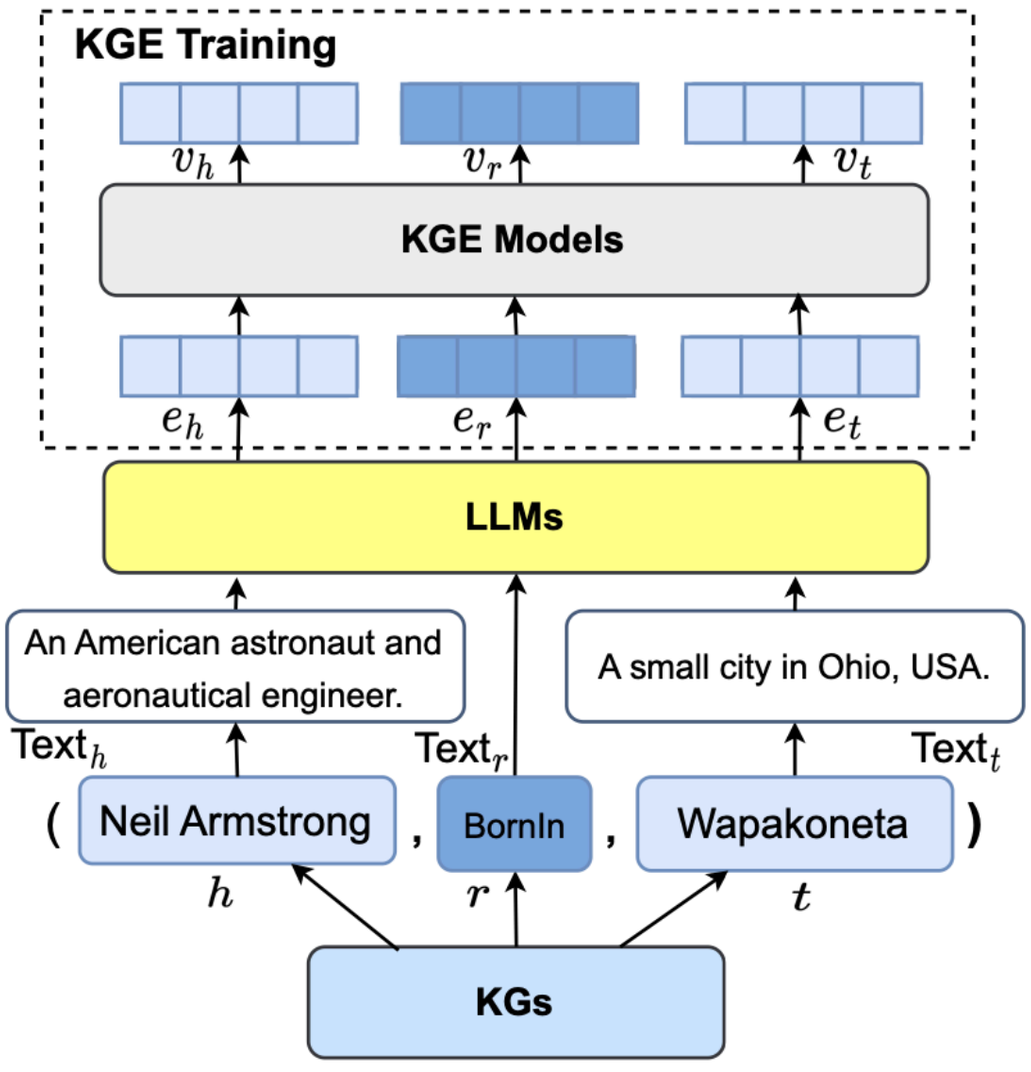
(a) TransE

- Note: one vector for each relation, additive modification only

Bordes et al. NIPS 2013. Translating Embeddings for Modeling Multi-relational Data

LLMs as encoder for KG Embeddings

- **Pretrain-KGE**: capture contextualized semantic meanings for nodes in a KG
  - Use LLMs to encode the text description of entities as the initial KG embeddings
  - Add an additional KGE model to further learn fine-grained representations.



Margin-based contrastive loss

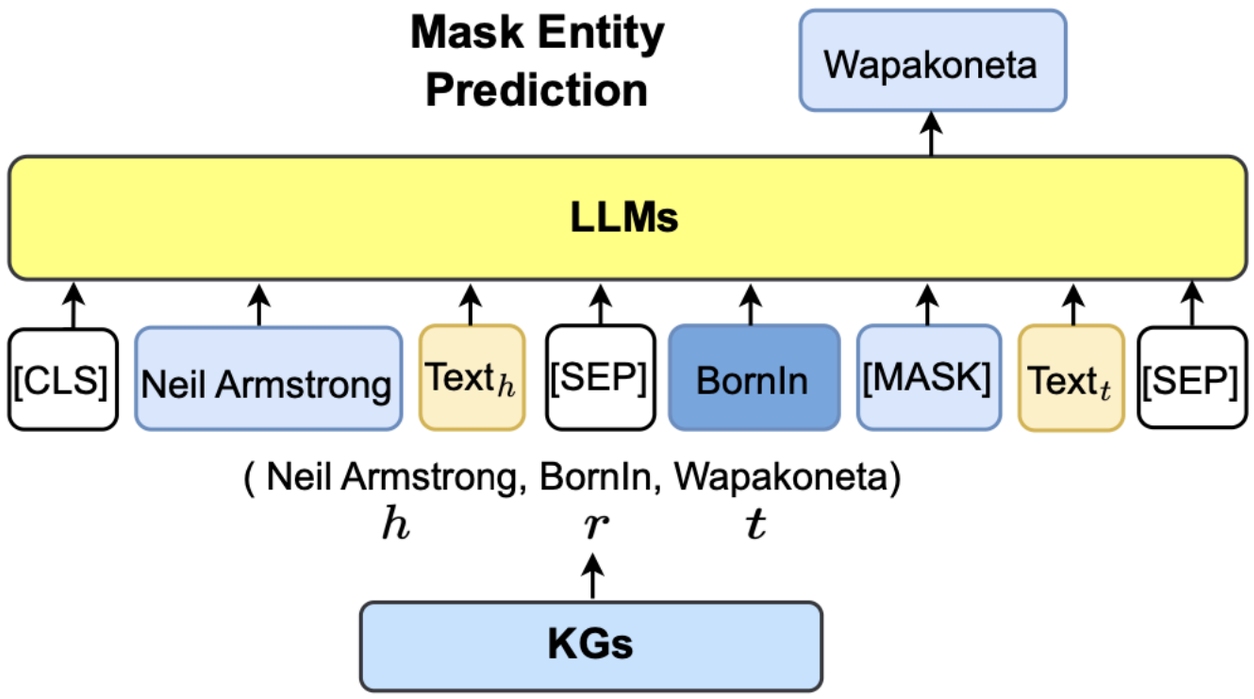
$$\mathcal{L} = [\gamma + f(v_h, v_r, v_t) - f(v'_h, v'_r, v'_t)],$$

$$e_h = \text{LLM}(\text{Text}_h), e_t = \text{LLM}(\text{Text}_t), e_r = \text{LLM}(\text{Text}_r),$$

Zhang et al. 2020. Pretrain-KGE: learning knowledge representation from pretrained language models

LLMs for Joint Text and KG Embeddings

- **kNN-KGE**:
  - treats the entities and relations as special tokens in the LLM.
  - During training, convert each relation triplet (*h, r, t*) into a sentence
  - Asks the model to predict the tail node given the head and relation.



$$P_{LLM}(t|h,r) = P(\texttt{[MASK]}=\texttt{t}|x,\Theta),$$

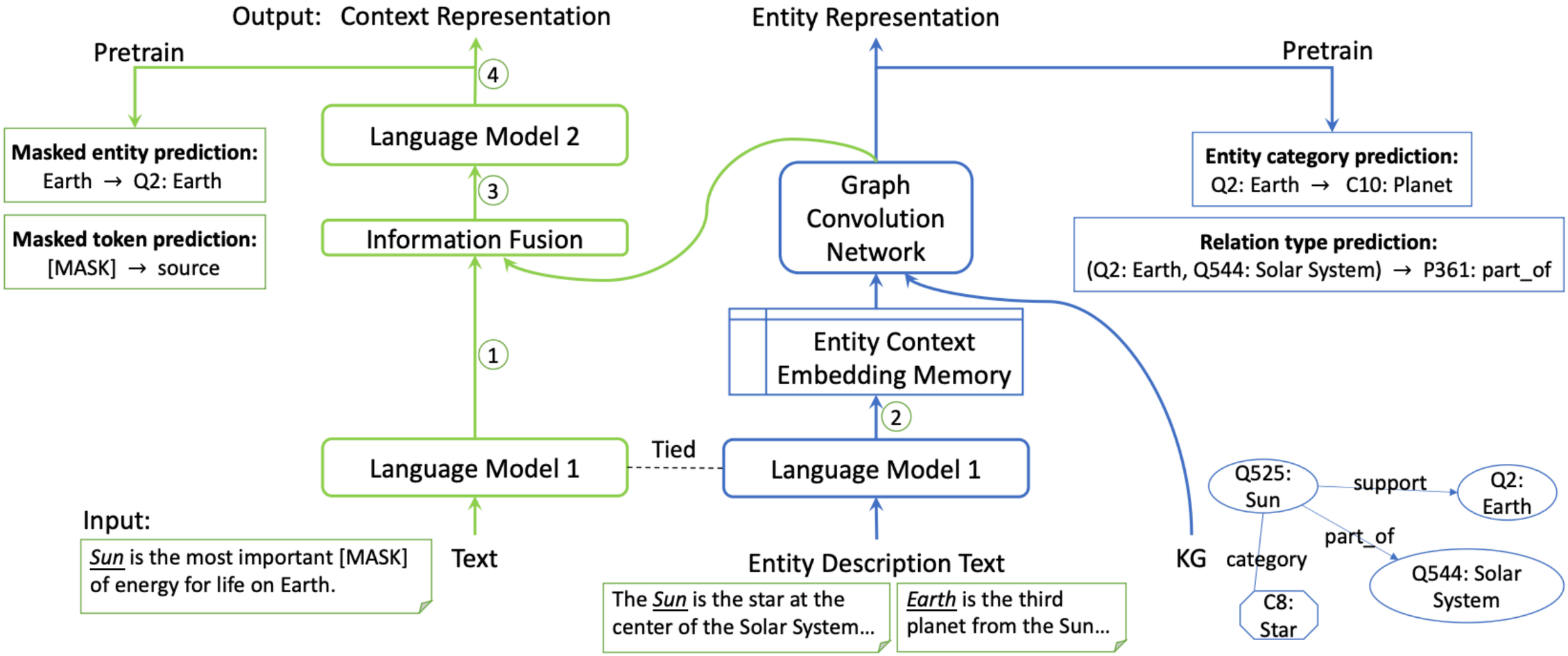
$$x = \texttt{[CLS]}\ h\ \text{Text}_h\ \texttt{[SEP]}\ r\ \texttt{[SEP]}\ \texttt{[MASK]}\ \text{Text}_t\ \texttt{[SEP]}$$

Zhang et al. 2020. Pretrain-KGE: learning knowledge representation from pretrained language models

LLM +GNN fusion for KG Embeddings

(JAKET, Yu et al. 2022)

- Instead of using only LLM as encoder, use **GNN to capture structure information**
- Self-supervised tasks: **Entity category prediction and relation type prediction**



Yu et al. 2022. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding

Task 2: KG completion — before LLMs

- **Information extraction (IE)**: extracting relation triplets from text
  - **Schema-based IE:** Pre-define a set of relations (a.k.a. schema) that we could extract for pairs of entities from text.
  - **OpenIE (schema free)**: predicts an open-set of relations (mostly based on linguistic syntax)

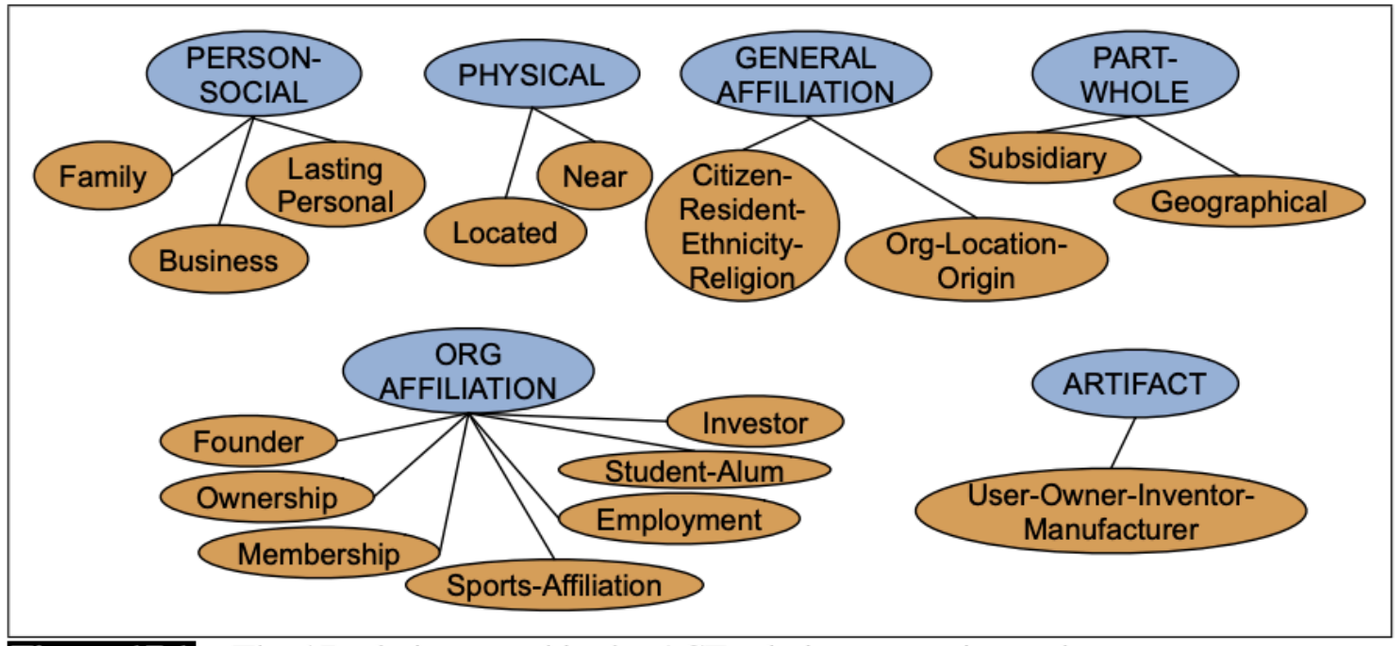


Figure 17.1 The 17 relations used in the ACE relation extraction task.

Relations	Types	Examples
Physical-Located	PER-GPE	**He** was in **Tennessee**
Part-Whole-Subsidiary	ORG-ORG	**XYZ**, the parent company of **ABC**
Person-Social-Family	PER-PER	**Yoko**'s husband **John**
Org-AFF-Founder	PER-ORG	**Steve Jobs**, co-founder of **Apple**...

Figure 17.2 Semantic relations with examples and the named entity types they involve.

Supervised Relation Extraction Baseline

- **Training**:
  - Labeled dataset: a KB triplet t=<e1, r, e2> on a sentence s
  - Supervised training of models (e.g., logistic regression, NNs)
- **Test**:
  - Find any pairs of entities in a sentence
  - Apply the relation classifier on all entity pairs

function FINDRELATIONS(*words*) **returns** *relations*

relations ← *nil*
entities ← FINDENTITIES(*words*)
forall **entity pairs** ⟨*e1*, *e2*⟩ **in** *entities* **do**
  if RELATED?(*e1*, *e2*)
    relations ← *relations*+CLASSIFYRELATION(*e1*, *e2*)

Distant Supervision for Relation Extraction (Mintz et al. 2009)

- **Motivation**: Supervised baseline is still limited to the labeled data size.
- Given an KB triplet *t=<e1, r, e2>*, extract all sentences (s_1, s_2, …s_N) that matches these two entities *<e1, e2>*, and use these texts to train the relation classifier

s_1 *[Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brothers' story.*

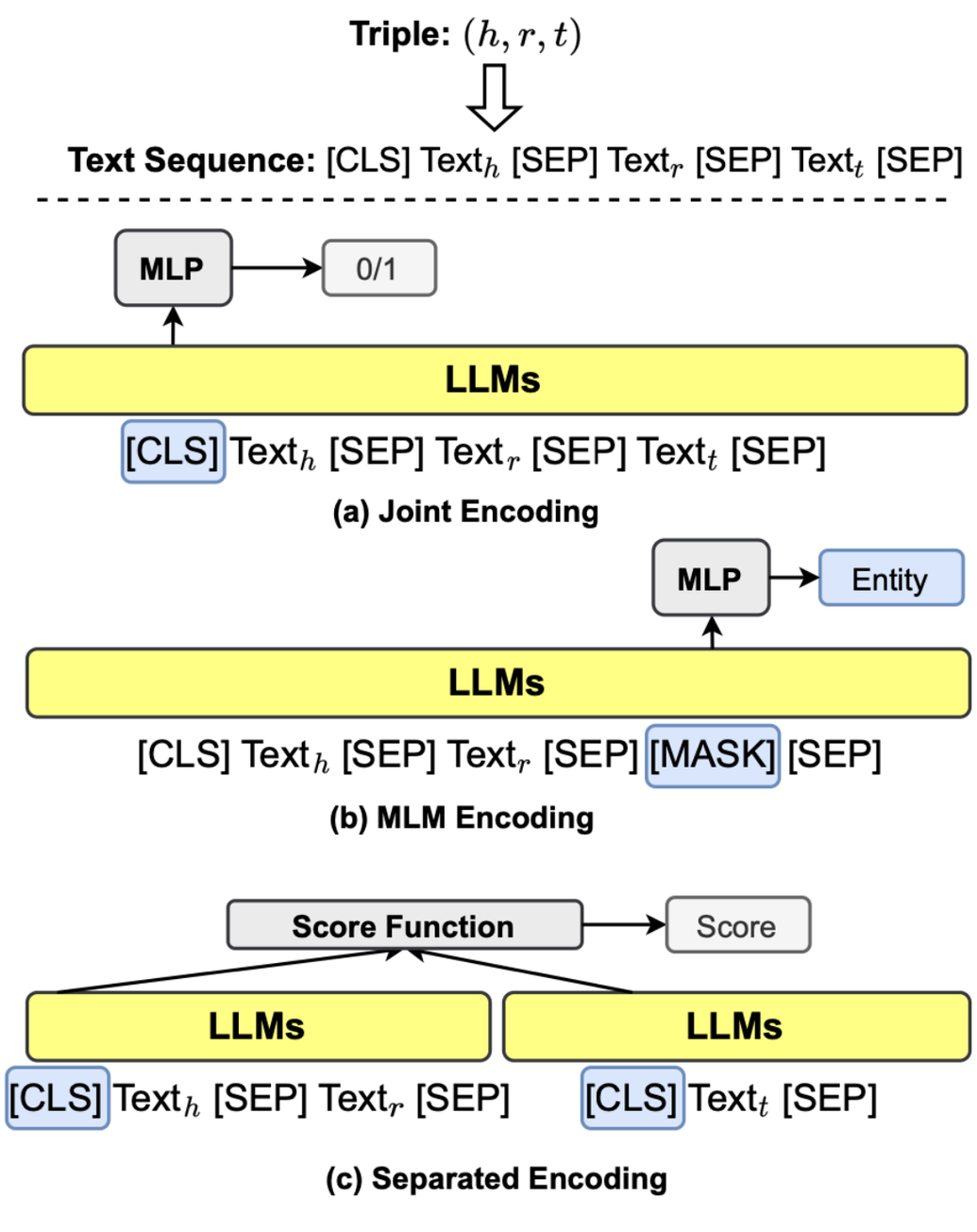
s_2 *Allison co-produced the Academy Award-winning [Saving Private Ryan], directed by [Steven Spielberg]...*

$$\text{classifier}(s_i, e_1, e_2) \rightarrow r, \quad \forall e_1 \in s_i \wedge e_2 \in s_i$$

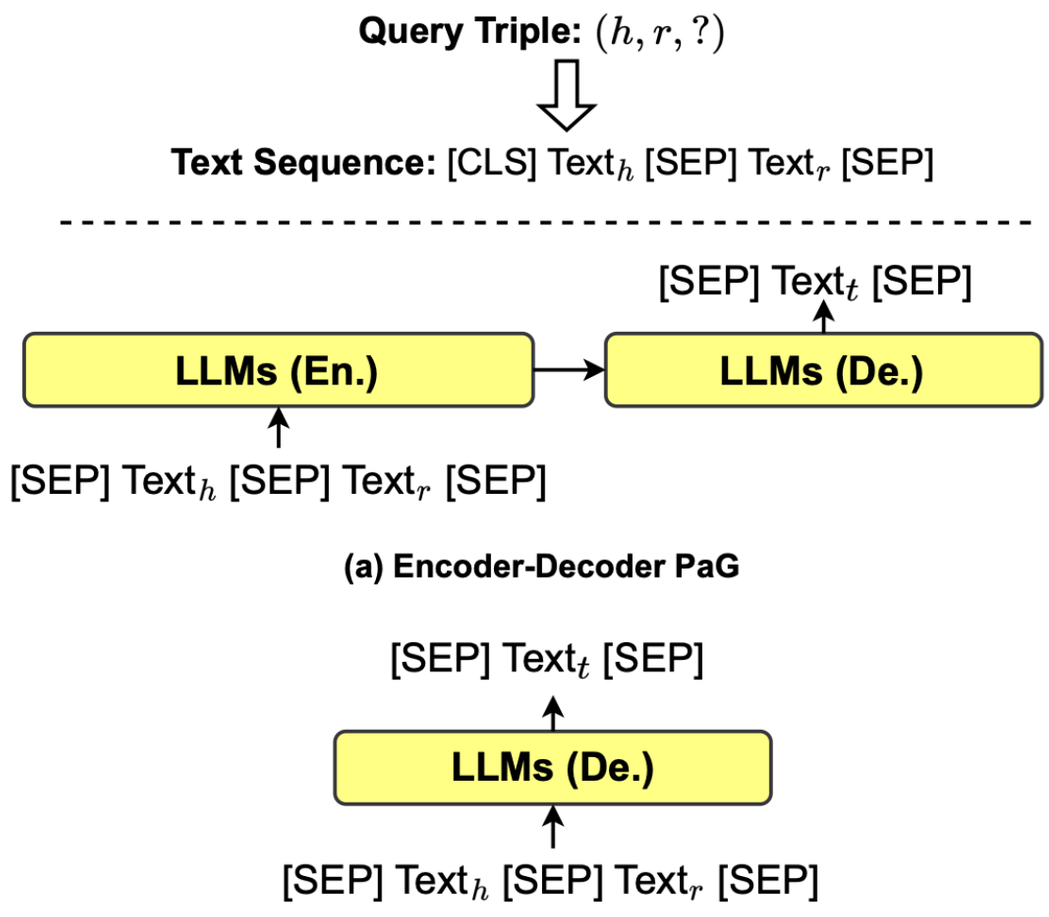
Mintz et al. 2009. Distant supervision for relation extraction without labeled data
Craven et al. 1998. Learning to extract symbolic knowledge from the World Wide Web

LLMs for KG Completion

- **Fine-tune** LLMs on fewer labeled data to predict more relation triplets

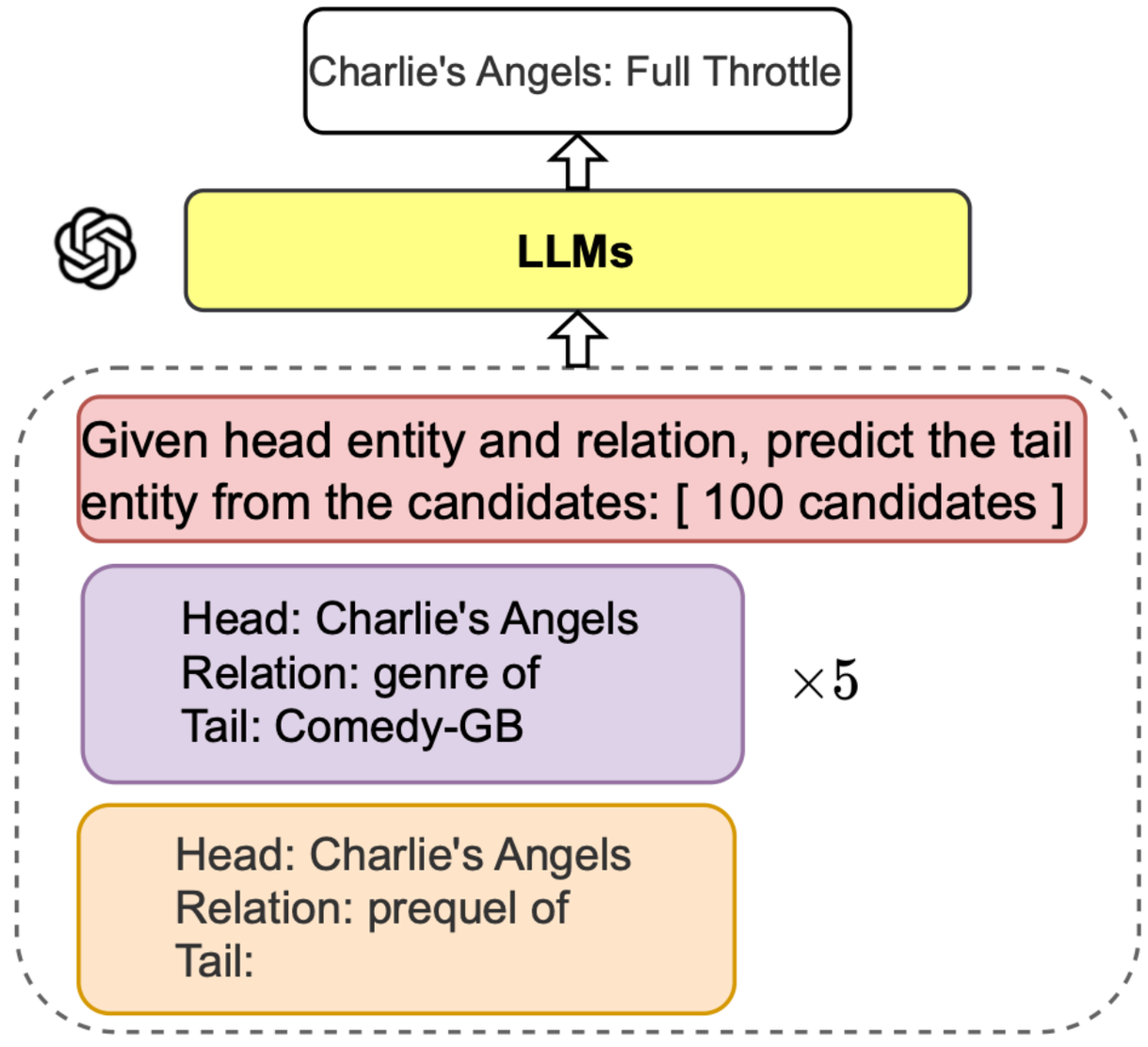


- **Prompt** LLMs w/o updating the model to predict relation triplets



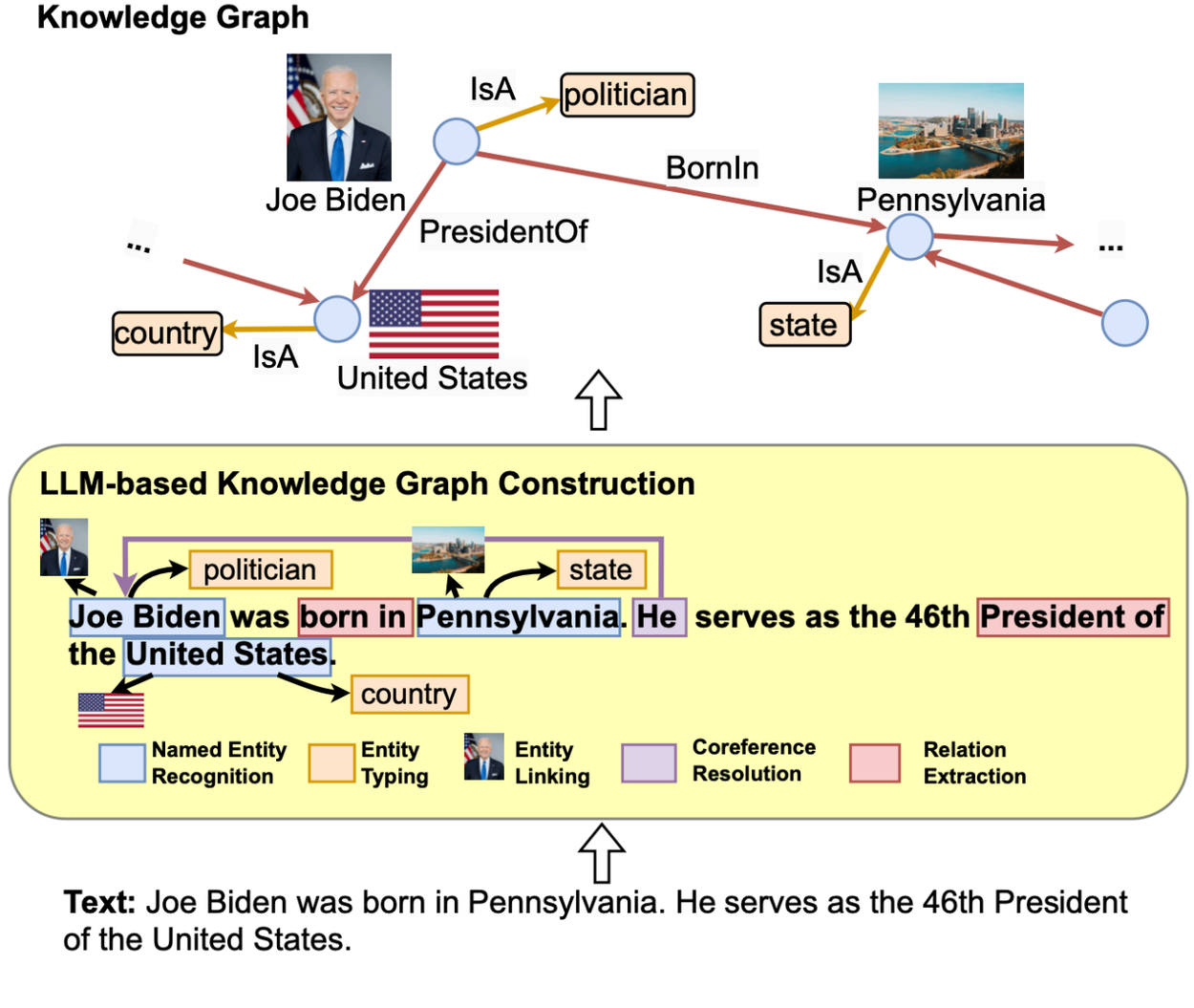
LLMs for KG Completion

- Given an existing KG, predict **missing relation triplets from more text data**
- **In-context prediction for a head entity**: Given a head node, add its 5 relation triplets in the prompt, and ask LLMs to predict the tail entity from the 100 candidates



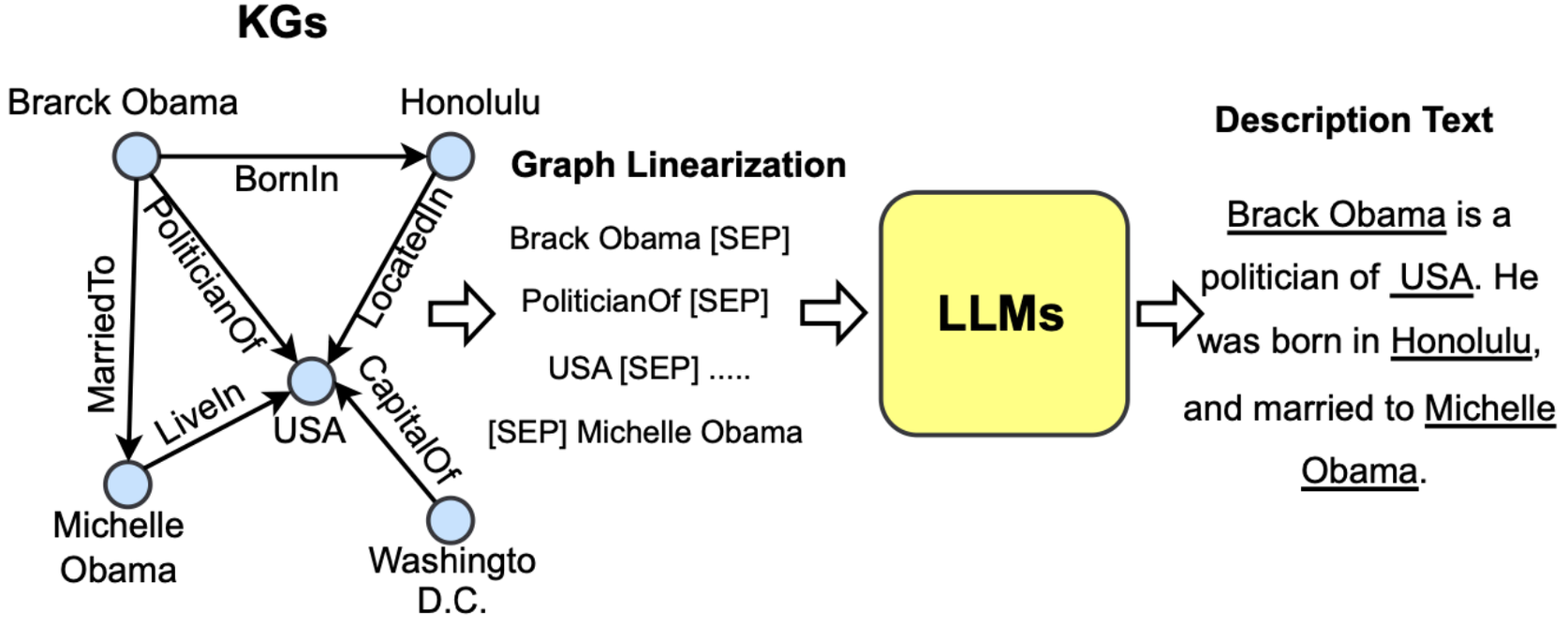
Task 3: LLMs for KG Construction

- Construct a KG from texts by a LLM from scratch
- **Predict all relation triplets in a sentence**: Prompt the LLM to predict a KG directly, which involves multiple tasks such as NER, Entity linking/typing, Coreference resolution, Relation prediction.
- **Before LLMs**, we build **a pipeline of multiple models** to construct KGs.



Task 4: LLM-augmented KG-to-text Generation

- Goal: generate high-quality texts that describe the input KG.
- Applications: include storytelling, KG-grounded dialogue.
- Challenge: collect large graph-text parallel data. So many studies use weakly-aligned graph-text pairs



Task 5: LLM-augmented KG for Question Answering

- **KGQA**: find answers to a question based on structured facts stored in KGs. The key challenge is to retrieve related facts and predict the answer based on the retrieved facts.

  - **Symbolic method (semantic parsing)**: convert a natural language query into a structured format (e.g., SQL, Python) to query the KG

  - **Neural symbolic method**: embed a natural language query and the knowledge base information into an embedding space, learn some integration modules to combine information, and make answer prediction

Knowledge Base Question Answering (KBQA)

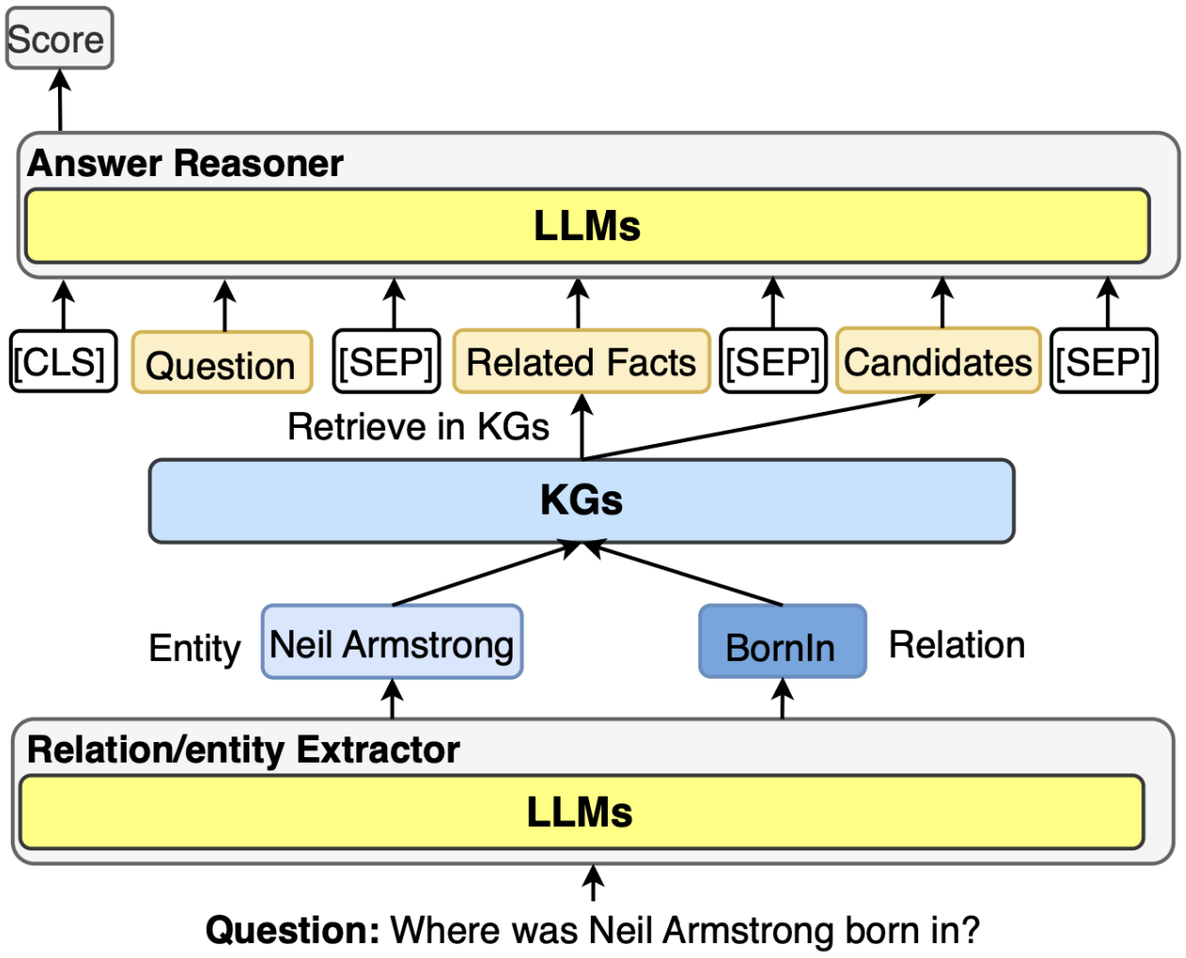
- Construct a KB from texts or other resources either manually or automatically

- **Symbolic method (semantic parsing)**: convert a natural language query into a structured format (e.g., SQL) to query the KB

- **Neural symbolic method**: embed a natural language query and the knowledge base information into an embedding space, learn some integration modules to combine information, and make answer prediction

LLM-augmented KG for Question Answering

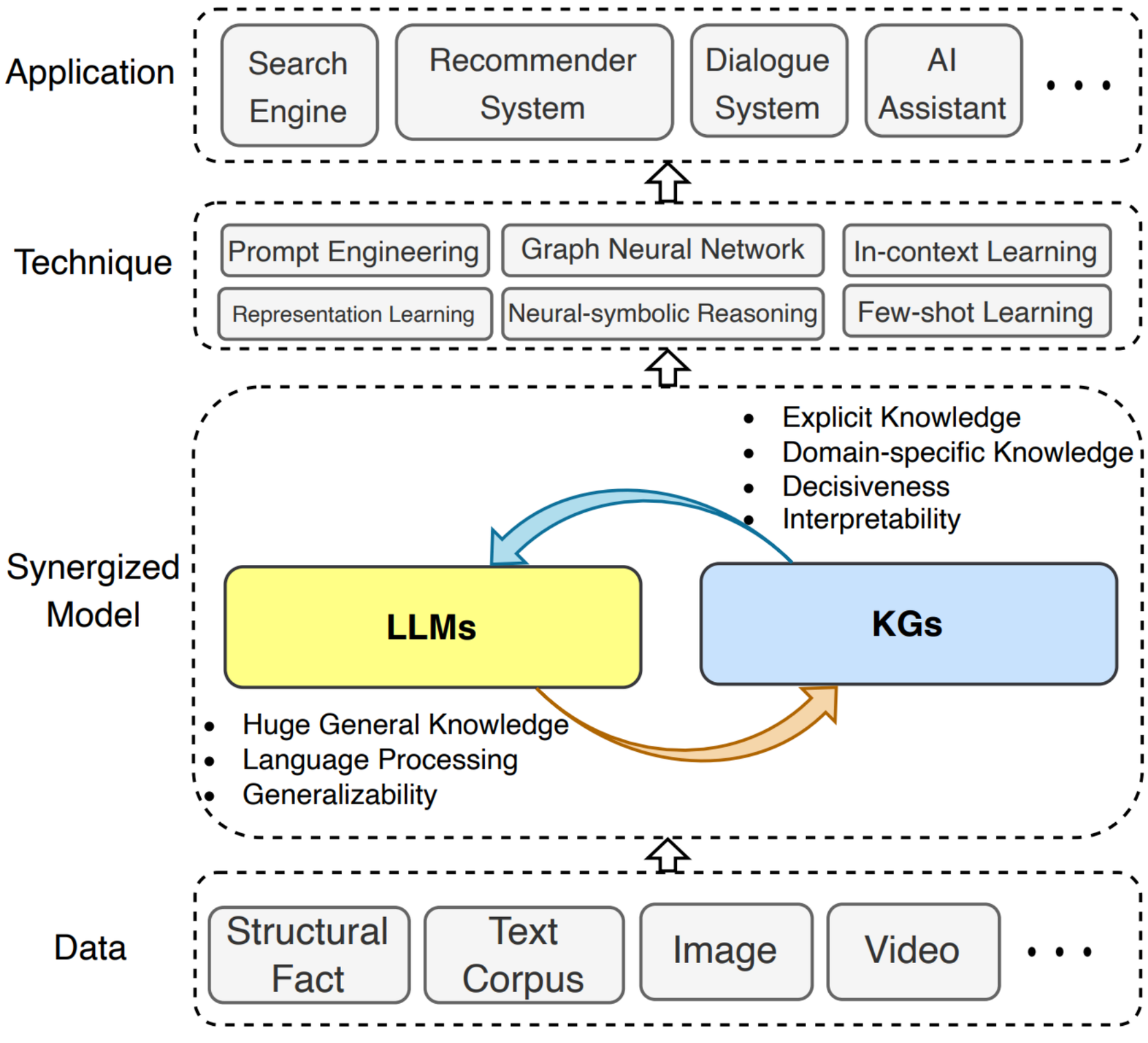
- **DEKCOR**: Use LLMs in two ways
  - Relation extractor: extract a head entity and a relation, and use them to retrieve related facts from KGs
  - Answer reasoner: concatenate a question with all related KG facts and a candidate answer as a sequence input to the LLM that outputs a score for this answer.
- **Limitation**: this assumes that we know all candidate answers beforehand, and also requires testing all candidate answers



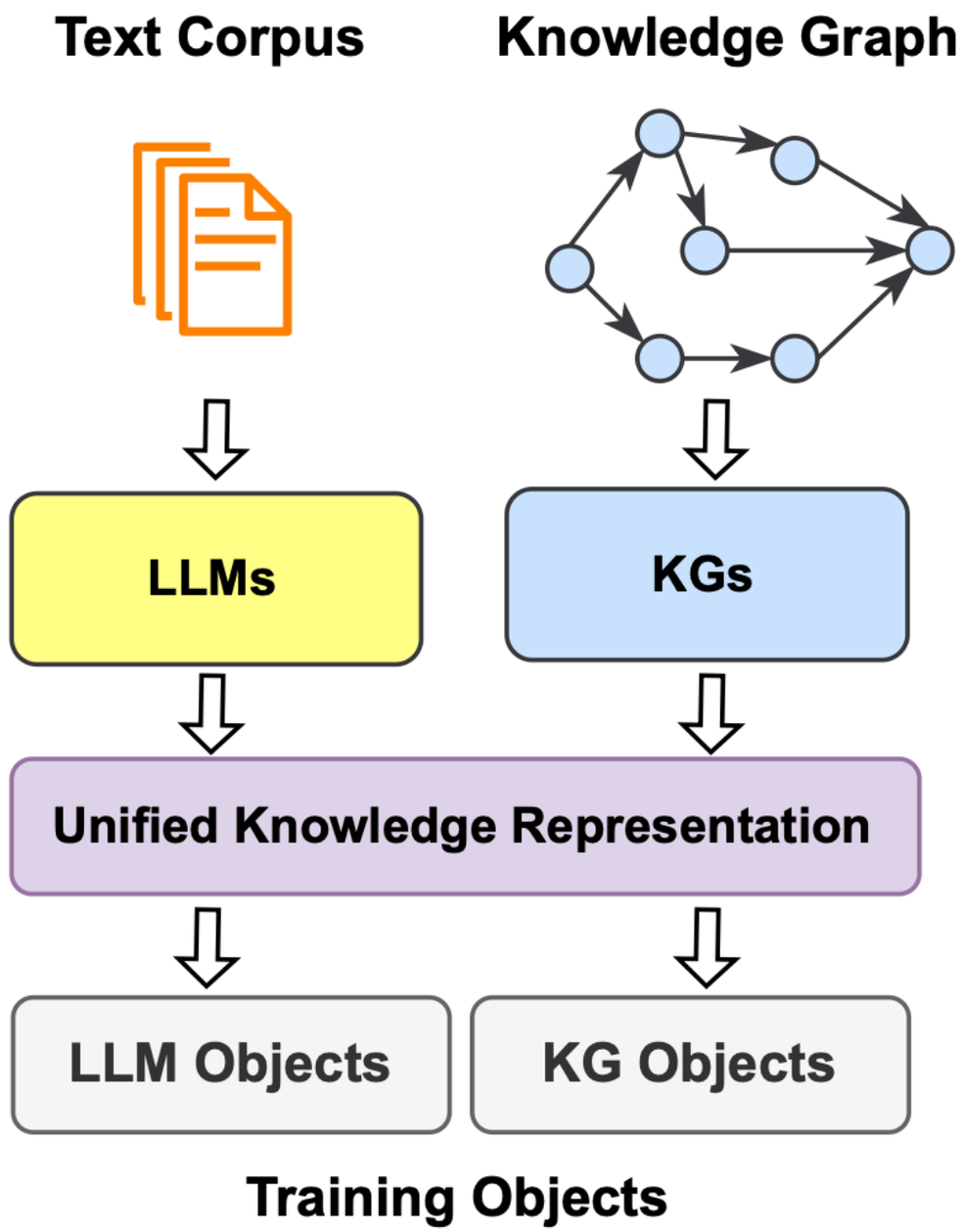
Xu et al. ACL 2021. Fusing context into knowledge graph for commonsense question answering

Unifying LLMs and KGs: Future directions

Synergized LLMs + KGs



Knowledge Representation of both unstructured and structured data

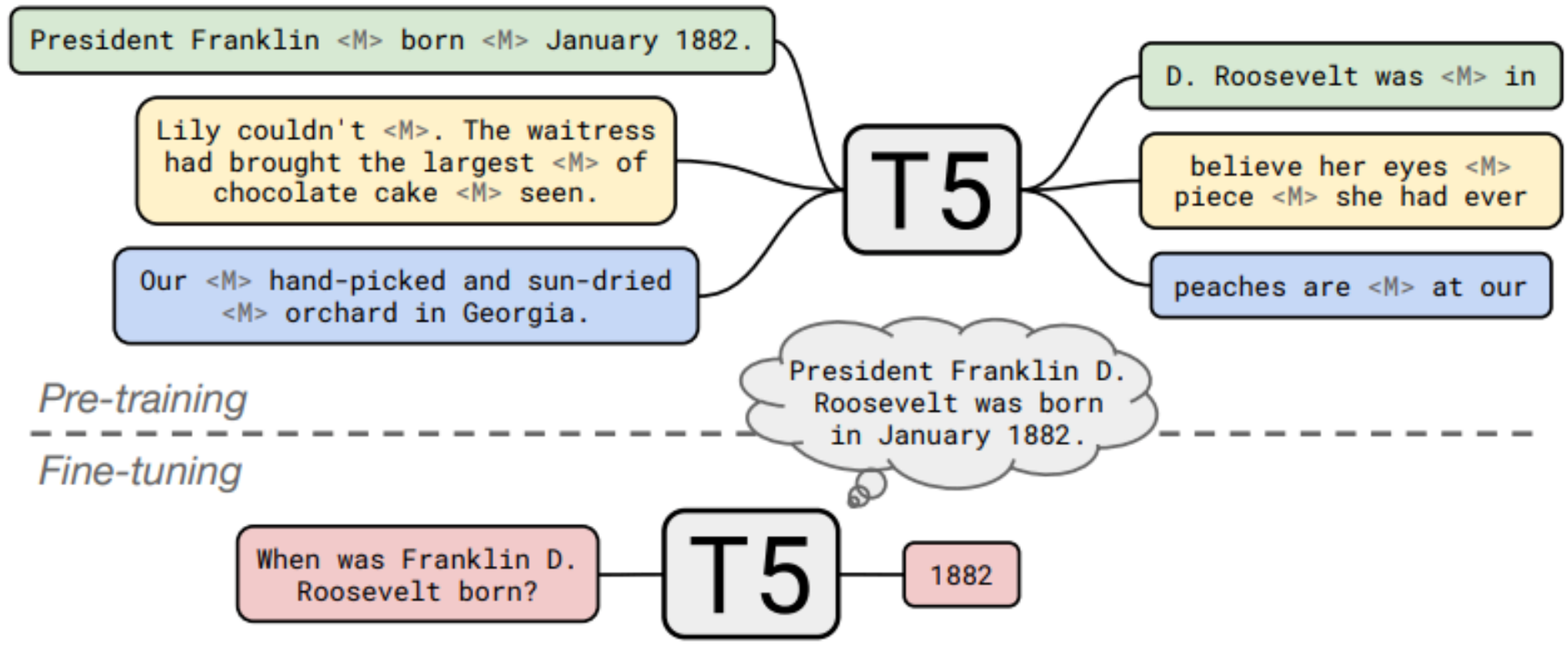


Comparison

- KBQA
  - Low coverage of knowledge
  - Faithful and interpretable
  - Dense structured
- TextQA
  - Wide coverage of knowledge
  - Misinformation
  - Massive raw texts
  - Enhanced with a text retrieval model
- LM-QA
  - Wide coverage of knowledge
  - Misinformation & out-dated information
  - Large model size
  - Black-box, not controllable

We need LLMs as backbone

- Pre-trained LLMs can generalize to understand even unseen questions/tasks.

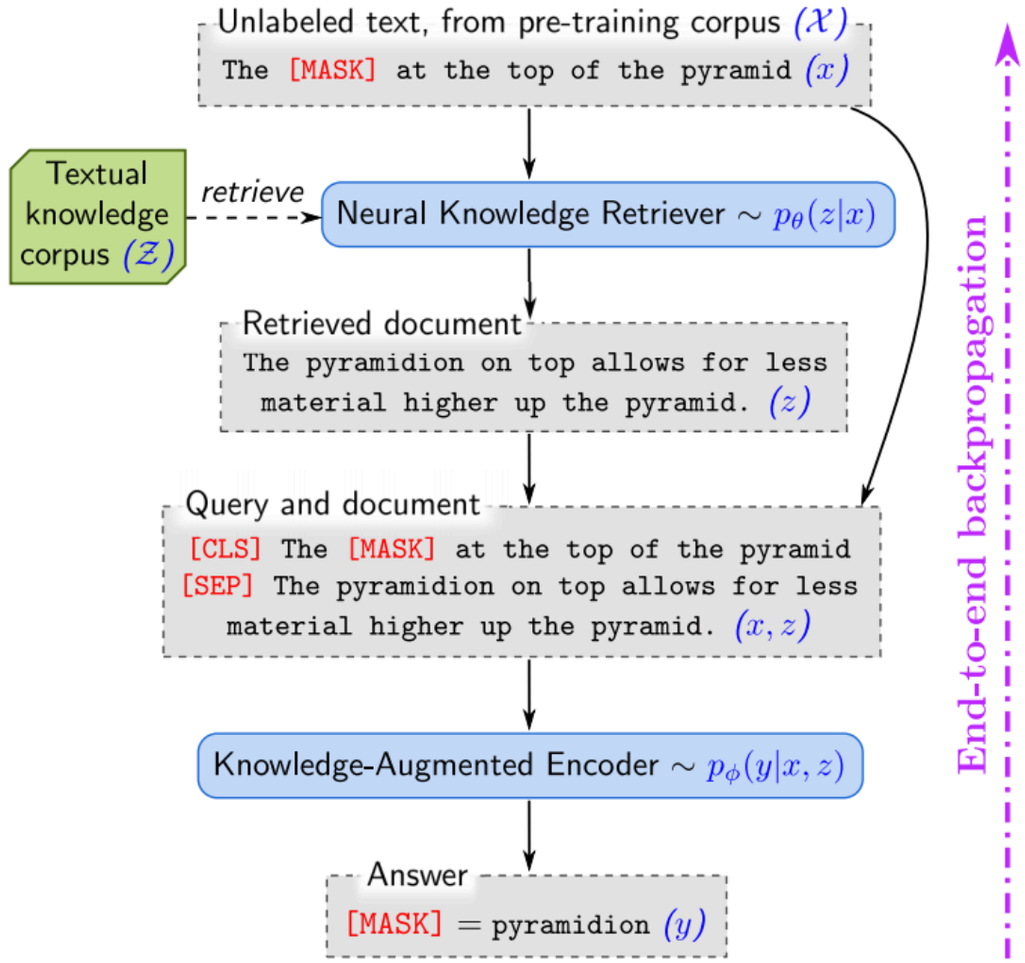


We need a retrieval component!

- For knowledge-intensive tasks like QA, nonparametric models (w/ retrieved context) outperform parametric models (w/o context) by a large margin.

- For example, REALM (Guu et al. 2020), RAG (Lewis et al. 2020) on the NaturalQuestion datasets.

Close-book T5	34.5
REALM	40.4
RAG	44.5



Future research directions in the era of LLMs

- **Representations**: How to encode structure data for LLMs?
  - Sequentialize KGs to text
  - Fuse KG embeddings to text embeddings
- **Decoding**: How to decode data that aligns with KGs?
  - Output a text sequence that follows a path in KG
  - Reuse LLM to generate KG directly
- **Alignment & Hallucination**: How to align LLM outputs to KG paths and detect hallucinations?
  - Add a validation step to check the correctness of LLM outputs against a KG
- **Multimodality**: How to encode multimodality data (images/videos) from KGs?
  - Align multimodal data representations into a shared embedding space
- **Interpretability**: How to edit (add/update/delete) knowledge in LLMs with KGs?
  - Understand what knowledge from KGs is stored in LLMs, and what are still missing or inconsistent.

Questions?