Knowledge Graphs and Language Models

Ernesto Jiménez-Ruiz
Lecturer in Artificial Intelligence

Agenda

- **Afternoon session**:
  - Theory: 13:15-14:45
  - Break 15 min
  - Hands-on: 15:00-16:30

Course Organization

- ✓ Introduction to Knowledge Graphs
  - ✓ Lab: Creation of a small knowledge graph and ontology.
- ✓ Reasoning and Querying with Knowledge Graphs
  - ✓ Lab: First steps with the SPARQL query language.
- ✓ Matching: KG-to-KG and CSV-to-KG
  - ✓ Lab: Creation of a (simple) matching system.
- 4. Knowledge Graphs and Language Models
  - – Lab: Ontology Embeddings with OWL2Vec*.

Turing Interest Group on KGs

Jeff Z. Pan

University of Edinburgh

Valentina Tamma

University of Liverpool

Ernesto Jiménez-Ruiz

City, University of London

Ian Horrocks

University of Oxford

Get in touch:

`https://www.turing.ac.uk/research/interest-groups/knowledge-graphs`

Organisers: `knowledgegraphs_tig@turing.ac.uk`

Previous meetups: January 26 (online), March 25 (Liverpool)

Hybrid Learning and Reasoning Systems

Motivation:

– Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.
- Limitations of KR systems: **maintenance** and **flexibility** in the inference. *e.g.*, Does $C(a)$ hold if?
  - A `and (`R `some` B`) subClassOf` C. $A(a)$, $B'(b)$, and $R(a, b)$.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.
- Limitations of KR systems: **maintenance** and **flexibility** in the inference. *e.g.*, Does $C(a)$ hold if?
  - A `and (`R `some` B`) subClassOf` C. $A(a)$, $B'(b)$, and $R(a, b)$.
- **Solution?** Hybrid Learning and Reasoning Systems.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods

† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). https://arxiv.org/abs/2102.11965

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods
- Overview of **patterns** for hybrid systems. †

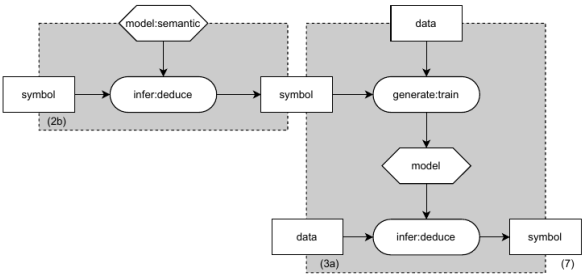
† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). https://arxiv.org/abs/2102.11965

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods
- Overview of **patterns** for hybrid systems. †
  - Focus on (1) **Ontology** (knowledge graph) **embeddings** (*e.g.*, OWL2Vec*) and (2) **PLMs**/**LLMs** as components for an hybrid system

† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). https://arxiv.org/abs/2102.11965

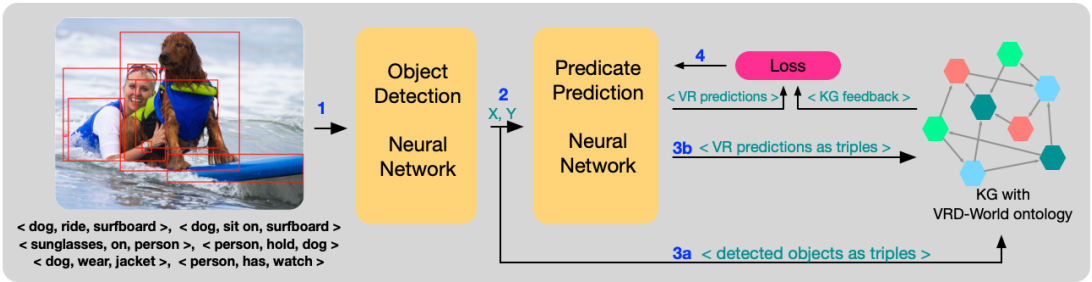
Learning with prior knowledge (i)



- Domain knowledge (*e.g.*, a KG) used to constraint search space during training.
- **Semantic loss function**: impact of the violation of the symbolic knowledge.

A semantic loss function for deep learning with symbolic knowledge. ICML 2018
Logic Tensor Networks. `https://github.com/logictensornetworks/logictensornetworks`

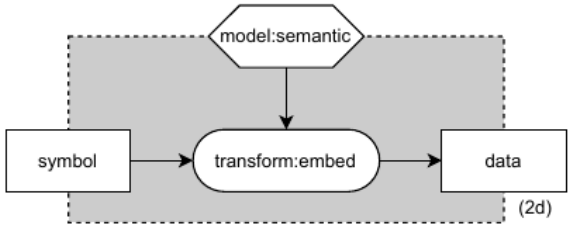
Learning with prior knowledge (ii)



– Penalisation of predictions that violate constrainst in the KG.

D. Herron et al. On the Potential of Logic and Reasoning in Neurosymbolic Systems using OWL-based Knowledge Graphs. Neurosymbolic Artificial Intelligence, 2024.

Knowledge graph embeddings



Symbols are transformed into vectors (*e.g.*, OWL2Vec*)

Knowledge Graph Embedding: A Survey of Approaches and Applications. TKDE 2017
OWL2Vec*: Embedding of OWL Ontologies. Machine Learning journal (2021)

Embeddings (definition)

– An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.

https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Embeddings (definition)

- An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.
- One-hot encoding also assign a vector to categories but...
  - ✗ has as many dimensions as categories $\rightarrow [0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ ...]$
  - ✗ vectors are not related to each other.

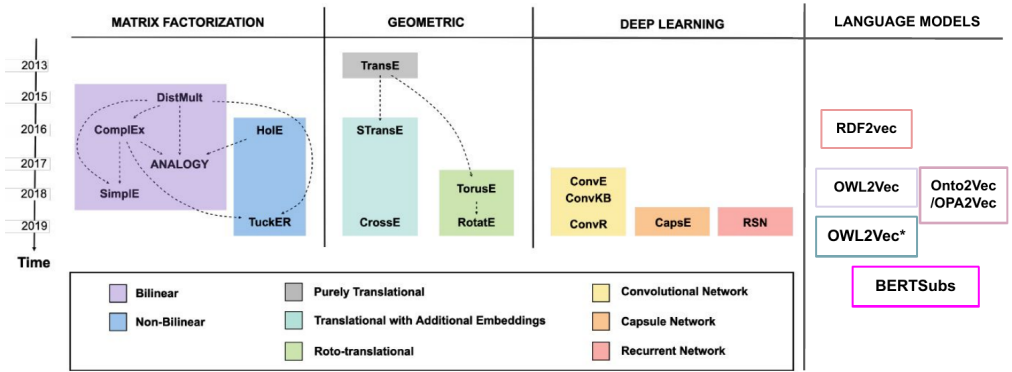
https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Embeddings (definition)

- An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.
- One-hot encoding also assign a vector to categories but...
  - ✗ has as many dimensions as categories $\rightarrow [0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ ...]$
  - ✗ vectors are not related to each other.
- Embeddings aim at creating **meaningful low-dimensional continuous vectors**.
  - dbr:london $\rightarrow [0.5\ \ 0.3\ \ 0.8\ \ 1.0\ \ 0.0\ ...]$

https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Knowledge graph embeddings (overview of approaches)



Incomplete list of approaches, adapted from: Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. TKDD 2021

Knowledge graph embeddings techniques (self-supervised)

KGE approaches (excluding those based on language models) typically:

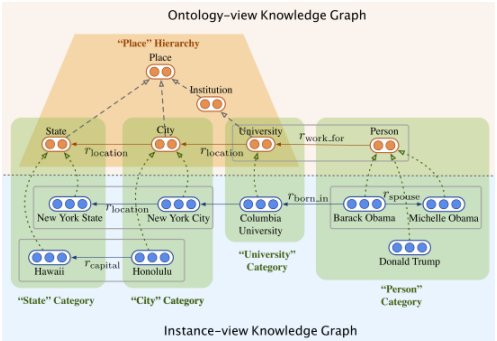
- Receive as input a set of **positive** (the ones in the KG) and **negative triples**.
- Include a **scoring function** that accepts as input the embedding of the elements of a triple (there is an initialization step).

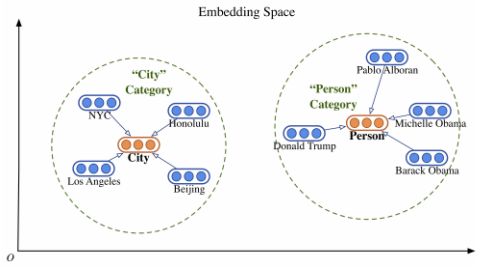
Knowledge graph embeddings techniques (self-supervised)

KGE approaches (excluding those based on language models) typically:

- Receive as input a set of **positive** (the ones in the KG) and **negative triples**.
- Include a **scoring function** that accepts as input the embedding of the elements of a triple (there is an initialization step).
- Learn embeddings so that the score for positive triples is maximized while the score for negative triples is minimized (*i.e.*, **loss function**).
- Compute **similar vectors** for similar nodes (*i.e.*, concepts/instances) and edges (*i.e.*, properties).

Knowledge graph embeddings (example)

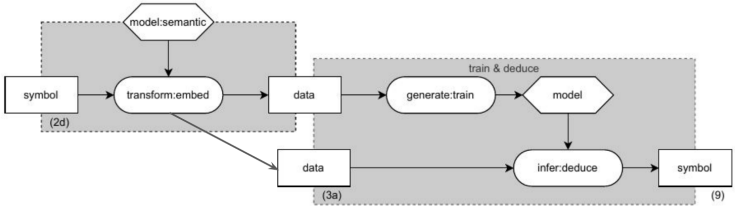




KG Embedding Systems exploit the neighbourhood of an entity to calculate its vector.

Example from: Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. KDD 2019.

Learning with (knowledge) embeddings (pattern)

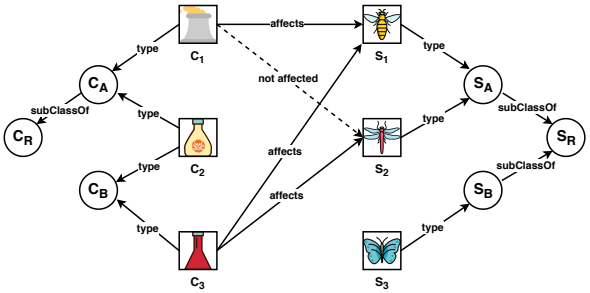


- Applying knowledge graph embeddings in a subsequent classification step.
- Key for zero-shot learning approaches

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.
Knowledge-aware Zero-Shot Learning: Survey and Perspective. arXiv:2103.00070. 2021

Learning with (knowledge) embeddings (example)

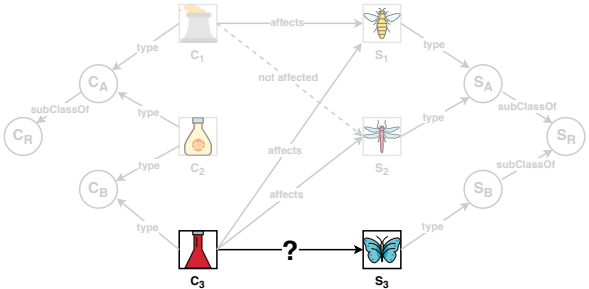
– **Prediction of adverse biological effects** of chemicals via KG embeddings.



Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

Learning with (knowledge) embeddings (example)

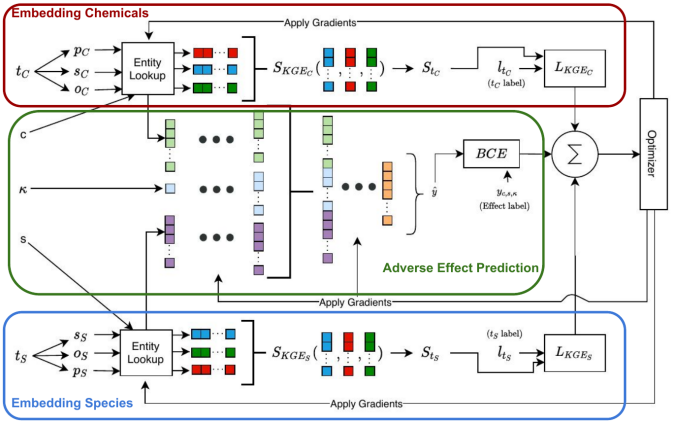
– **Prediction of adverse biological effects** of chemicals via KG embeddings.



KGE are critical for unseen chemicals and species. Also useful for explainability.

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

Learning with (knowledge) embeddings (example)

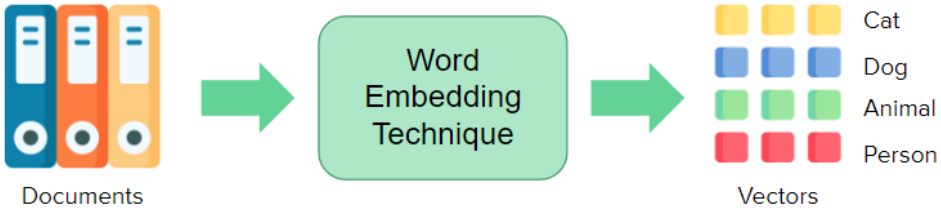


Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

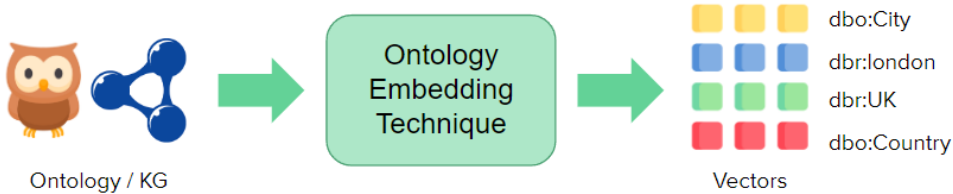
KGs and Language Models: KG Embeddings and beyond

Preliminaries: embeddings

Embedding techniques



Embedding techniques



Word Embedding Techniques (non-contextual)

- **One-hot** embedding.
- Frequency-based Embeddings:
  - **Co-occurrence Matrix**.
  - **TF-IDF** (Term Frequency-Inverse Document Frequency)
  - **GloVe** (Global Vectors for Word Representation)
- **Prediction-based** Embeddings :
  - **Word2Vec** (uses Neural Networks)
  - **FastText** (extends Word2Vec with Subword Information)

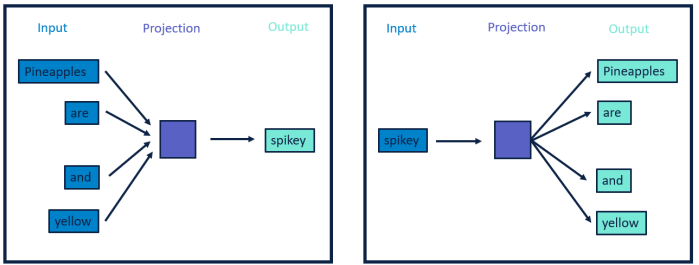
Word2Vec (i)

- **Word2Vec** is a two-layer neural network
- Each unique word is assigned a (**low-dimensional and dense**) vector.
- Two architectural designs: the Continuous Bag of Words (**CBOW**) Model and the Continuous **Skip-Gram** Model.
- Vectors are learned (via an objective function) to capture the **semantic** meaning of the words and **proximity** to other words

Tomas Mikolov,et al. Efficient Estimation of Word Representations in Vector Space. 2013

Word2Vec (ii)

- CBOW: learns to predict a word given its neighboring words.
- Skip-gram: learns to predict neighboring words given a target word.



https://swimm.io/learn/large-language-models/what-is-word2vec-and-how-does-it-work\

Word2Vec (iii)

Limitations:

- Non contextual embeddings (*bank* vs *river bank*)
- Bias on the embeddings.
- Problem with "out of the vocabulary" words.
- Subwords do not necessarily have similar embeddings (*'end'* and *'endless'*).

Word2Vec (iii)

Limitations:

- Non contextual embeddings (*bank* vs *river bank*)
  - Solved by next generation of (L)LMs.
- Bias on the embeddings.
- Problem with "out of the vocabulary" words.
- Subwords do not necessarily have similar embeddings (*'end'* and *'endless'*).

Word2Vec (iii)

Limitations:

- Non contextual embeddings (*bank* vs *river bank*)
  - Solved by next generation of (L)LMs.
- Bias on the embeddings.
  - Still a challenge.
- Problem with "out of the vocabulary" words.
- Subwords do not necessarily have similar embeddings (*'end'* and *'endless'*).

Word2Vec (iii)

Limitations:

- Non contextual embeddings (*bank* vs *river bank*)
  - Solved by next generation of (L)LMs.
- Bias on the embeddings.
  - Still a challenge.
- Problem with "out of the vocabulary" words.
  - Minimised in FastText and in LLMs.
- Subwords do not necessarily have similar embeddings (*'end'* and *'endless'*).

Word2Vec (iii)

Limitations:

- Non contextual embeddings (*bank* vs *river bank*)
  - Solved by next generation of (L)LMs.
- Bias on the embeddings.
  - Still a challenge.
- Problem with "out of the vocabulary" words.
  - Minimised in FastText and in LLMs.
- Subwords do not necessarily have similar embeddings (*'end'* and *'endless'*).
  - Minimised in FastText and LLMs.

Preliminaries: Contextual embeddings

Transformer-based models (i)

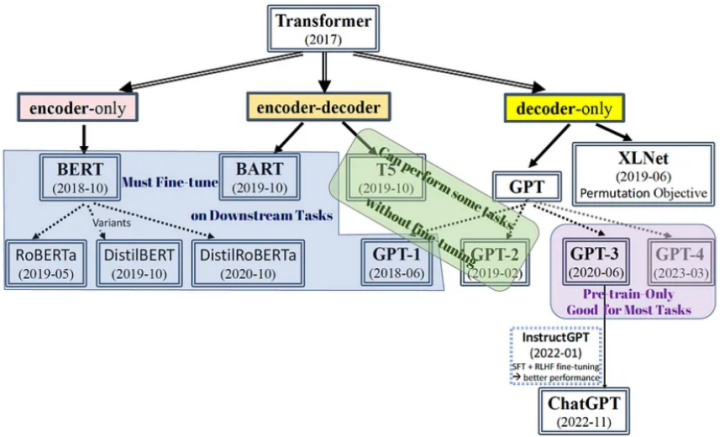
- **Learn contextual embeddings** for each of the words (*i.e.*, a word will have different vectors depending on the context).
- **Pre-trained on (very) large corpora** of text data using unsupervised learning objectives.
- Some require to be **fine-tuned** on specific downstream tasks with labelled data to achieve high performance.
- Have achieved **state-of-the-art performance** on various NLP tasks.

Attention Is All You Need: `https://arxiv.org/pdf/1706.03762.pdf`

Transformer-based models (ii)

- **Three types of models**:
  - Transformer encoders (aka Autoencoders models).
  - Transformer decoders (aka Autoregressive models).
  - Transformer Encoder-Decoder (aka seq2seq models)
- Some examples:
  - **BERT** (Bidirectional Encoder Representations from Transformers),
  - **GPT** (Generative Pre-trained Transformer),
  - **T5** (Text-To-Text Transfer Transformer), and
  - **XLNet**.

Transformers-based models: Taxonomy



https://medium.com/the-modern-scientist/an-in-depth-look-at-the-transformer-based-models-22e5f5d17b6b

Transformer-decoder or AR: GPT-based models

- Used by well-known **GPT models**.
- Use the context to predict the **likelihood of the next word**.
- A **deep neural network** (billions of parameters) is trained to model these conditional distributions.
- Only trained to encode a **uni-directional context** (either forward or backward).
- Shown impressive potential for **text generation**.
- Harder to fine-tune, but ready to be used in **zero-shot/few-shot** via prompting (scenarios).

AR - Transformer-decoder

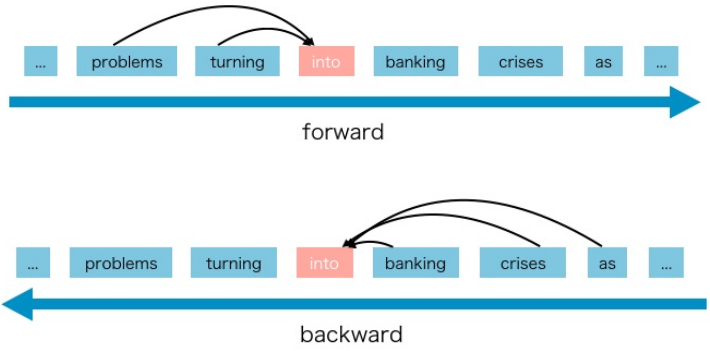


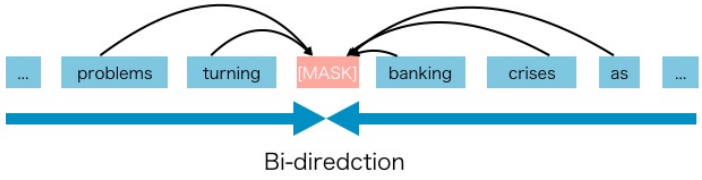
Image from: `https://aman.ai/primers/ai/autoregressive-vs-autoencoder-models/`

Transformer-encoder or AE: BERT-based models (i)

- Used in well-known **BERT-based models** (Bidirectional Encoder Representations from Transformers)
- **Deep neural network** architecture with million of parameters.
- Pre-training aims to reconstruct the original data from corrupted input (*e.g.*, symbol `[MASK]`) $\rightarrow$ **masked language model**.
- Shown impressive performance (after **fine-tuning**) in downstream **text classifications** tasks: spam detection, sentiment analysis, topic categorisation, language detection.

Transformer-encoder or AE: BERT-based models (ii)

– BERT can capture bidirectional context.



– The problem is the introduction of artificial symbols like `[MASK]`.

Image from: `https://aman.ai/primers/ai/autoregressive-vs-autoencoder-models/`

Encoder-decoder/seq2seq models

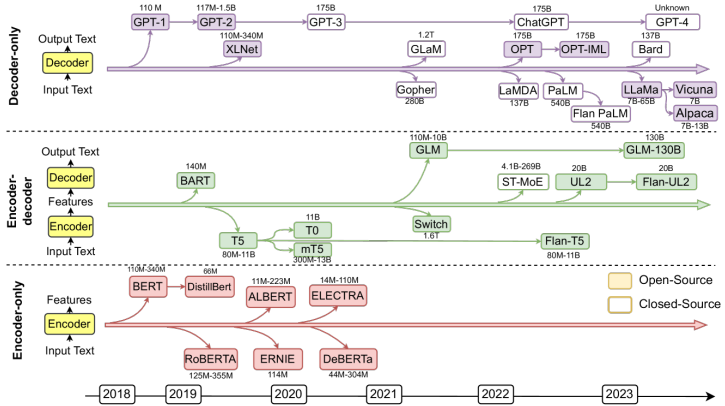
- Use both an encoder and a decoder.
- Each task is considered a sequence to sequence conversion/generation.
- Typically used for tasks that require both content understanding (encoder) and generation (decoder). For example, translation.

LLM Variants (August 2023)

Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models.

`https://arxiv.org/abs/2308.14149`

LLM Variants (January 2024)



Shirui Pan, et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 2024.

LLMs and KGs: Opportunities and Challenges

Explicit vs. Parametric Knowledge

- **Explicit knowledge:** unstructured knowledge such as text, images and videos; and structured knowledge (*i.e.*, symbolic knowledge) such as knowledge graphs.
- **Parametric knowledge:** refer to the implicit knowledge encoded into the language models' internal parameters (*e.g.*, weights of the neural network).

A key research line is how to transform parametric knowledge into symbolic knowledge. Transformer models can contain **billions of parameters**.

Debate points

- LLMs have shown to generalize from large-scale text corpora.
- LLMs provide plausible answers but not necessarily factually correct.
- LLMs have problems with long-tail knowledge.
- LLMs issues with respect to bias, fairness, copyright violation and misinformation. Hard to “forget” such toxic information from LLMs.
- LLM explainability and interpretability of their predictions.

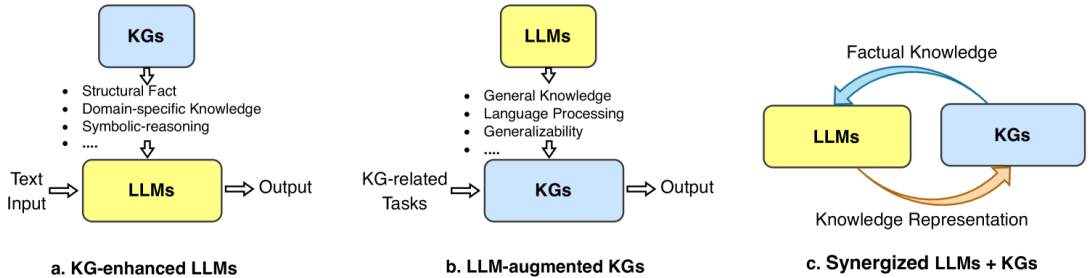
Jeff Pan et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. TGDK 2023.

Opportunities: LLMs & KGs (i)

- **Explicit-Knowledge-First:** “LLMs will enable, advance, and simplify crucial steps in the knowledge engineering pipeline so much as to enable Ks at unprecedented scale, quality, and utility.”
- **Parametric-Knowledge-First:** “KGs will improve, ground, and verify LLM generations so as to significantly increase reliability and trust in LLM usage.”

Jeff Pan et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. TGDK 2023.

Opportunities: LLMs & KGs (ii)



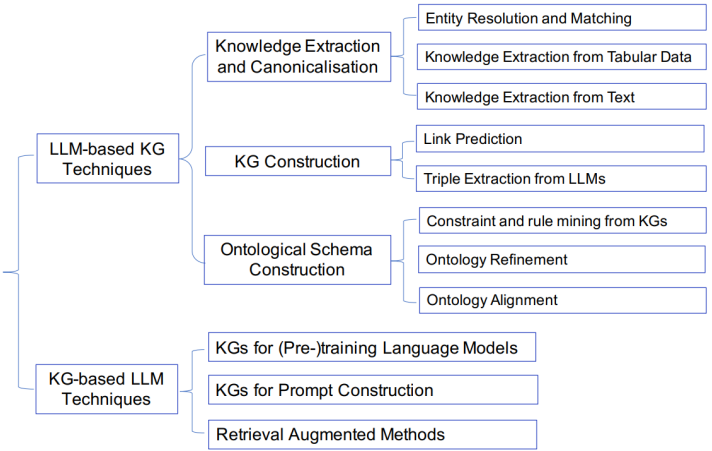
Shirui Pan, et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 2024.

Opportunities: LLMs & KGs (iii)

1. LLMs for KGs: Knowledge Extraction and Canonicalisation
2. LLMs for KGs: KG Construction
3. LLMs for KGs: Ontological Schema Construction
4. KGs for LLMs: Training and Augmenting LLMs

Jeff Pan et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. TGDK 2023.

Opportunities: LLMs & KGs (iv)



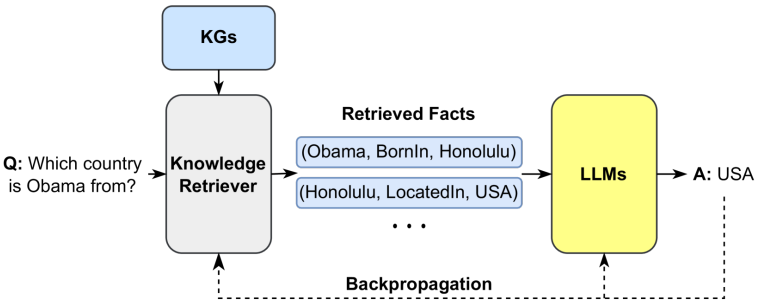
Jeff Pan et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. TGDK 2023.

KGs for LLMs

KG-enhanced LLM Inference

KG-enhanced LLM Inference

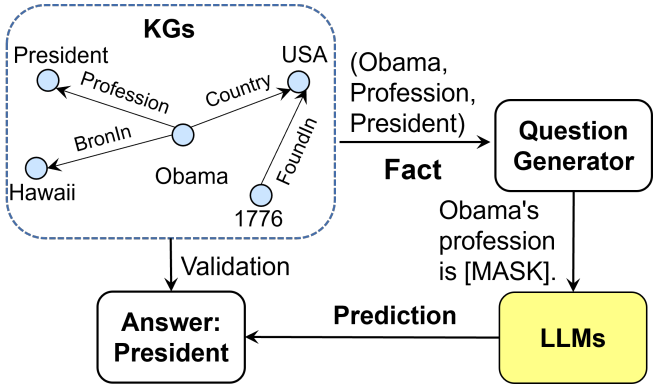
Good to provide the LLMs with fresh/up-to-date facts (without the need of retraining).



Shirui Pan, et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 2024.

KG-enhanced LLM interpretability

KG-enhanced LLM interpretability: Probing

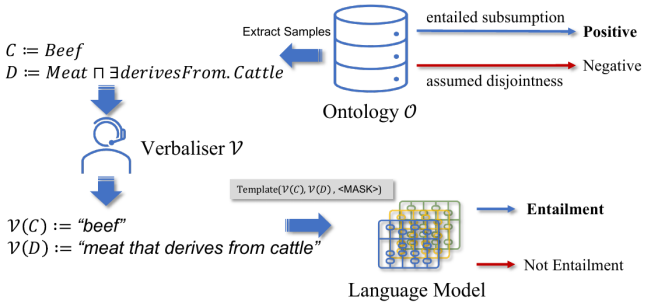


Shirui Pan, et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 2024.

KG-enhanced LLM interpretability: Ontology Inference Probing

OntoLAMA: Language Model Analysis for Ontology Inferencing

– To what extent **PLMs infer ontology semantics**? (*e.g.*, $Beef \sqsubseteq Meat$)



Y. He et al. Language Model Analysis for Ontology Subsumption Inference. ACL findings 2023. https://arxiv.org/abs/2302.06761

KG-enhanced LLM interpretability: Ontology Inference Probing

- To what extent **PLMs infer ontology semantics**? (*e.g.*, $C \sqsubseteq D$)
- Natural Language Inference (NLI) for $C \sqsubseteq D$:
  - Premise: "x is a C" (*e.g.*, "x is a $Beef$")
  - Hypothesis: "x is a D" (*e.g.*, "x is a $Meat$")
- Templates *(Template(C, D, <MASK>))*:
  - x is a C, is x a D? <Mask>
  - Is it [a/an] C? <MASK>, it is [a/an] D (used in paper)

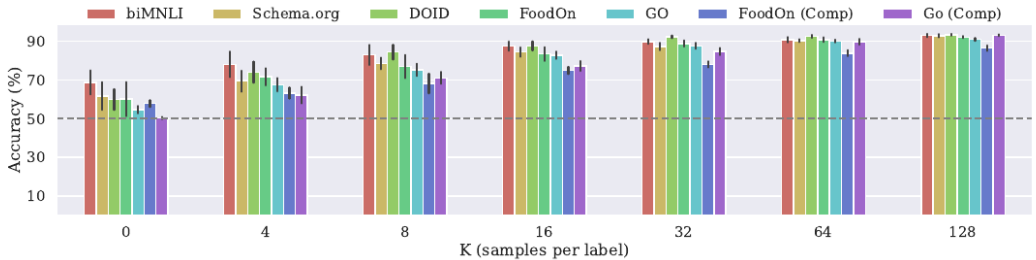
(*) C and D represent labels for atomic concepts or the verbalization for complex concepts.

Y. He et al. Language Model Analysis for Ontology Subsumption Inference. ACL findings 2023. https://arxiv.org/abs/2302.06761

KG-enhanced LLM interpretability: Ontology Inference Probing

OntoLAMA: Language Model Analysis for Ontology Inferencing

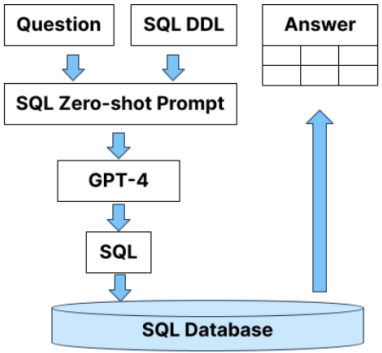
- To what extent **PLMs infer ontology semantics**? (*e.g.*, $Beef \sqsubseteq Meat$)
- **Prompt-based Inference** using RoBERTa in a **K-shot** setting.

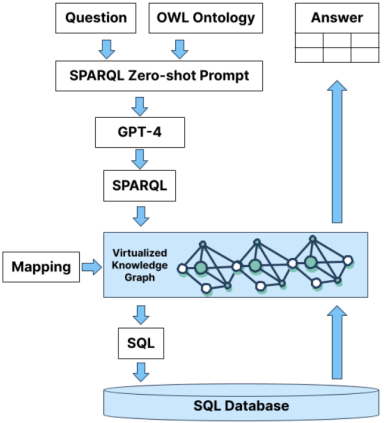


Y. He et al. Language Model Analysis for Ontology Subsumption Inference. ACL findings 2023. https://arxiv.org/abs/2302.06761

KG-enhanced LLM Question Answering

KGs and LLMs for Question Answering (i)





Juan Sequeda, Dean Allemang, Bryon Jacob: A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases. 2023 https://arxiv.org/abs/2311.07509

KGs and LLMs for Question Answering (ii)

	w/o KG (SQL)	w/ KG (SPARQL)	Improvement
All Questions	16.7%	54.2%	37.5%
Low Question/Low Schema	25.5%	71.1%	45.6%
High Question/Low Schema	37.4%	66.9%	29.5%
Low Question/High Schema	0%	35.7%	35.7%
High Question/High Schema	0%	38.5%	38.5%

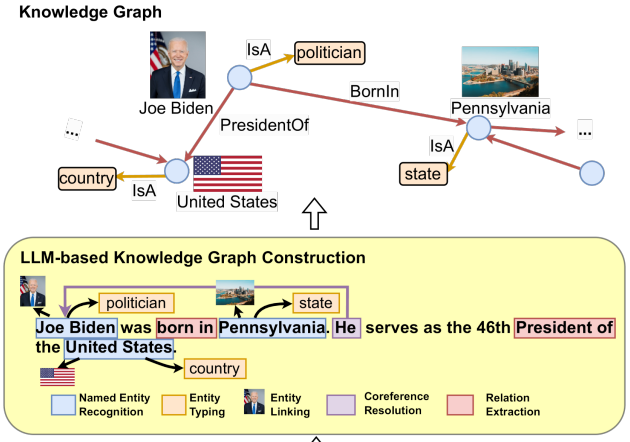
Juan Sequeda, Dean Allemang, Bryon Jacob: A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases. 2023 `https://arxiv.org/abs/2311.07509`

Keynote Turing IG on KGs: `https://github.com/turing-knowledge-graphs/meet-ups/blob/main/agenda-7th-meetup.md`

LLMs for KGs

LLM-Enhanced KG Extraction

Knowledge Extraction from Text



Text: Joe Biden was born in Pennsylvania. He serves as the 46th President of the United States.

Knowledge Extraction from Tabular Data

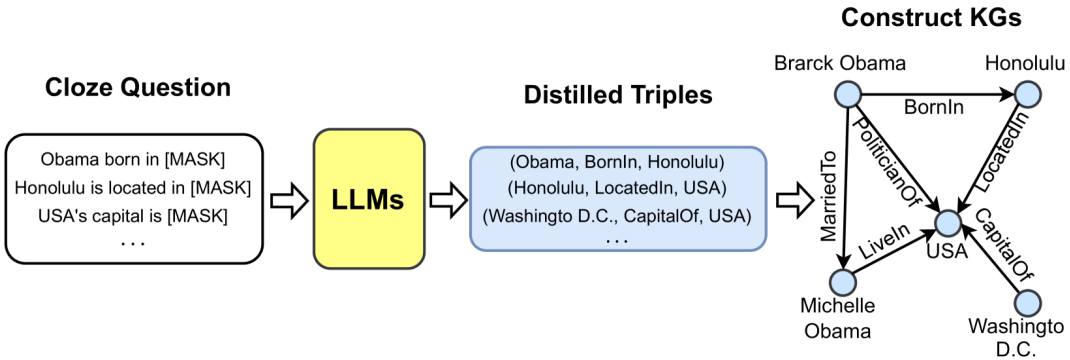
Answer the question based on the task below. If the question cannot be answered using the information provided answer with "I don't know".
Task: Classify the columns of a given table with only one of the following classes that are separated with comma: description of event, description of restaurant, postal code, region of address …
Table: Column 1 || Column 2 || Column 3 || Column 4 \n
Friends Pizza ||2525|| Cash Visa MasterCard || 7:30 AM\n
Class:

name of restaurant, postal code, payment accepted, time

Keti Korini, Christian Bizer. Column Type Annotation using ChatGPT. VLDB Workshops 2023

LLM-Enhanced KG Completion

LLM-Enhanced KG Completion

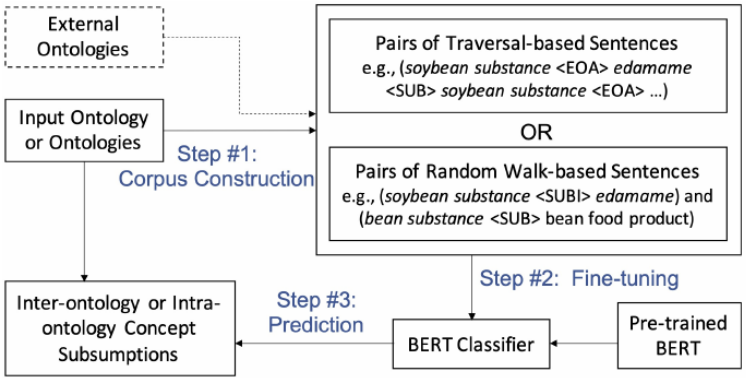


Similar to the probing case but to obtain fresh triples.

Shirui Pan, et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE TKDE, 2024.

BERTSubs embeddings for ontology subsumption

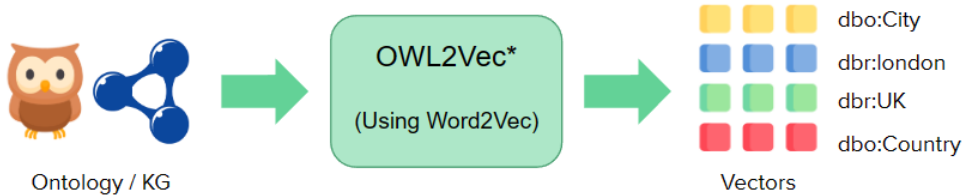
BERTSubs fine-tunes a pre-trained BERT model for ontology subsumption prediction.



J. Chen et al. Contextual Semantic Embeddings for Ontology Subsumption Prediction. World Wide Web Journal 2023

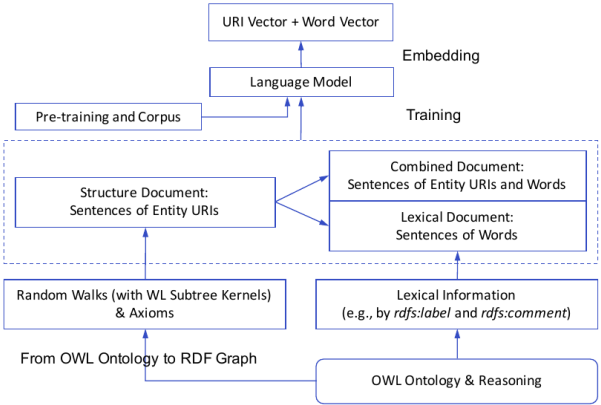
Language Models for KG Embeddings

OWL2Vec*: ontology embeddings with Word2Vec (i)



OWL2Vec*: ontology embeddings with Word2Vec (ii)

- **projects** the ontology into a graph,
- **walks** the graph,
- creates a **corpus of sentences** according to the walking strategies, and
- generates **embeddings** from that corpus using **Word2Vec**.



OWL2Vec*: Embedding of OWL Ontologies. Machine Learning journal 2021.

OWL2Vec*: ontology embeddings with Word2Vec (iii)

Projection: Approximation of an OWL 2 ontology into an RDF graph.

Axiom of Condition 1	Axiom or Triple(s) of Condition 2	Projected Triple(s)
$A \sqsubseteq \square r.D$ or $\square r.D \sqsubseteq A$	$D \equiv B \mid B_1 \sqcup ... \sqcup B_n \mid B_1 \sqcap ... \sqcap B_n$	$\langle A, r, B \rangle$ or $\langle A, r, B_i \rangle$ for $i \in 1, ..., n$
$\exists r.\top \sqsubseteq A$ (domain)	$\top \sqsubseteq \forall r.B$ (range)	
$A \sqsubseteq \exists r.\{b\}$	$B(b)$	
$r \sqsubseteq r'$	$\langle A, r', B \rangle$ has been projected	
$r' \equiv r^-$	$\langle B, r', A \rangle$ has been projected	
$s_1 \circ ... \circ s_n \sqsubseteq r$	$\langle A, s_1, C_1 \rangle ... \langle C_n, s_n, B \rangle$ have been projected	
$B \sqsubseteq A$	–	$\langle B, rdfs{:}subClassOf, A \rangle$ $\langle A, rdfs{:}subClassOf^-, B \rangle$
$A(a)$	–	$\langle a, rdf{:}type, A \rangle$ $\langle A, rdf{:}type^-, a \rangle$
$r(a, b)$	–	$\langle a, r, b \rangle$

$\square$ is one of: $\geq$, $\leq$, $=$, $\exists$, $\forall$. A, B, B_i and C_i are atomic concepts (classes), s_i, r and r' are roles (object properties), r^- is the inverse of a relation r, a and b are individuals, $\top$ is the top concept.

OWL2Vec*: ontology embeddings with Word2Vec (iv)

Strategies to generate sentences:

- Random walks
- Weisfeiler Lehman (WL) kernel, which assign identifiers to subgraphs and includes them into the walk.

Structure Document Sentences

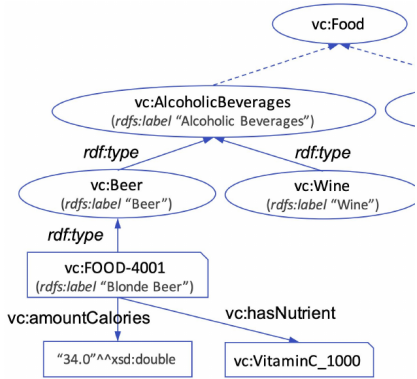
(vc:Beer, rdf:type, vc:FOOD-4001, vc:hasNutrient, vc:VitaminC_1000)

Lexical Document Sentences

("beer", "type", "blonde", "beer", "has", "nutrient", "vitamin", "c")

Combined Document Sentences

(vc:FOOD-4001, "has", "nutrient", "vitamin", "c")
OR
("blonde", "beer", "has", "nutrient", vc:VitaminC_1000)



OWL2Vec*: ontology embeddings with Word2Vec (v)

- OWL2Vec* relies on the **Word2vec** as neural **language model**.
- Word2vec learns **embeddings** for all the elements in the documents (*i.e.*, both **words** and **URIs**)

OWL2Vec*: ontology embeddings with Word2Vec (v)

- OWL2Vec* relies on the **Word2vec** as neural **language model**.
- Word2vec learns **embeddings** for all the elements in the documents (*i.e.*, both **words** and **URIs**)
- The embeddings of the ontology entities can be calculated via their **URI embedding** or via the **word embeddings** of their labels.
  - The URI `vc:FOOD-4001` (Blonde Beer) has a vector.
  - As well as the words ‘‘`blonde`’’ and ‘‘`beer`’’.

Rule Mining Using LLMs

A Rule Mining Case with ChatGPT 3.5: uncle

You

Please list three rules that could help to infer one person's uncle. And output it in as datalog rules.

A Rule Mining Case with ChatGPT 3.5: uncle

You

Please list three rules that could help to infer one person's uncle. And output it in as datalog rules.

1. **Uncle Rule based on Sibling Relationship:**

datalog Copy code

```
uncle(X, Y) :- sibling(X, Z), parent(Z, Y), male(X).
```

A Rule Mining Case with ChatGPT 3.5: uncle

You

Please list three rules that could help to infer one person's uncle. And output it in as datalog rules.

1. **Uncle Rule based on Brother's Child Relationship:**

datalog Copy code

```
uncle(X, Y) :- brother(X, Z), parent(Z, Y), male(X).
```

A Rule Mining Case with ChatGPT 3.5: uncle

You

Please list three rules that could help to infer one person's uncle. And output it in as datalog rules.

1. **Uncle Rule based on Marriage**:

datalog Copy code

```
uncle(X, Y) :- married(X, Z), sibling(Z, W), parent(W, Y), male(X).
```

A Rule Mining Case with ChatGPT 3.5: chemical exposure

You

Please list rules that could help to infer chemicals affecting species. And output it in as datalog rules. Use short predicate names.

A Rule Mining Case with ChatGPT 3.5: chemical exposure

You

Please list rules that could help to infer chemicals affecting species. And output it in as datalog rules. Use short predicate names.

1. **Chemical Exposure Rule based on Direct Observation:**

datalog — Copy code

```
exposure(S, C) :- observed(S, C).
```

A Rule Mining Case with ChatGPT 3.5: chemical exposure

You

Please list rules that could help to infer chemicals affecting species. And output it in as datalog rules. Use short predicate names.

1. **Chemical Exposure Rule based on Habitat:**

datalog Copy code

```
exposure(S, C) :- inhabits(S, H), hazard(C, H).
```

A Rule Mining Case with ChatGPT 3.5: chemical exposure

You

Please list rules that could help to infer chemicals affecting species. And output it in as datalog rules. Use short predicate names.

1. **Chemical Exposure Rule based on Food Chain**:

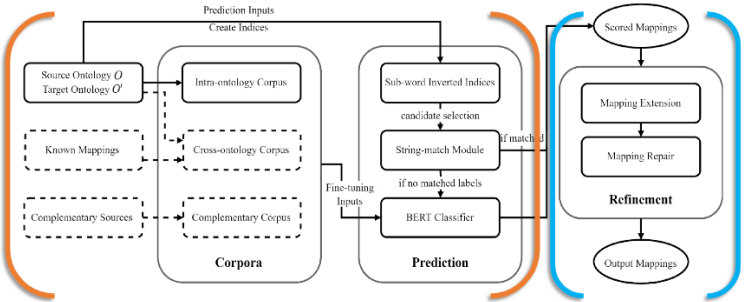
datalog — Copy code

```
exposure(S, C) :- consumes(S, P), exposure(P, C).
```

LLMs for Ontology Alignment

BertMap: Bert-based Ontology Alignment

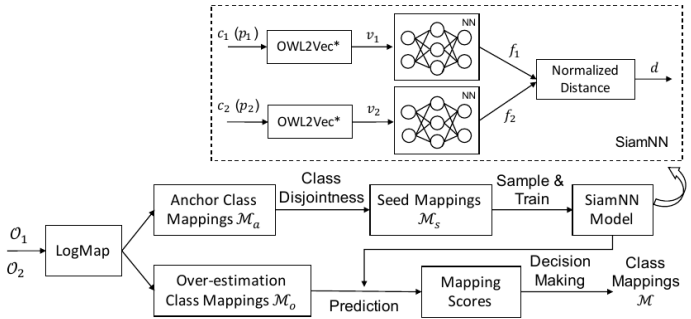
BertMap: fine-tunes BERT with (1) ontology entity synonyms and non-synonyms (unsupervised), and optionally with (2) example mappings (semi-supervised).



Yuan He et al: BERTMap: A BERT-Based Ontology Alignment System. AAAI 2022: 5684-5691.

OWL2Vec*: application to ontology alignment

- LogMap + OWL2Vec* + ML = LogMap-ML
- Self-supervised ontology matching



J. Chen, E. Jiménez-Ruiz et al. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. ESWC 2021

OWL2Vec*: application to ontology alignment (ii)

Method	Mappings #	Precision	Recall	F1 Score
LogMap^{anc}	139	0.892	0.479	0.629
LogMap^{anc}-ML	157	0.917	0.555	0.691
LogMap	190	0.842	0.618	0.713
LogMap-ML	190	0.881	0.645	0.745
LogMap^{oaei}	198	0.843	0.645	0.731
LogMap^{oaei}-ML	197	0.875	0.665	0.756
AML^{oaei}	220	0.827	0.703	0.760
AML^{oaei}-ML	222	0.842	0.723	0.778

– Results for the OAEI Conference track

– The architecture can be integrated with other OA systems (*e.g.*, AML).

LLMs for Ontology Alignment (i)

- **OntoLAMA/probing** setting applied to inter-ontology subsumptions
- Key: successfully include **context in the prompts**.

James Boyd. MSc Data Science @ City. Investigating OWL Ontology Alignment With Language Models.

LLMs for Ontology Alignment (i)

- **OntoLAMA/probing** setting applied to inter-ontology subsumptions
- Key: successfully include **context in the prompts**.
- Potential templates:
  - *The source entity is* C*, the target entity is* D*. Are the concepts equivalent? <MASK>*

James Boyd. MSc Data Science @ City. Investigating OWL Ontology Alignment With Language Models.

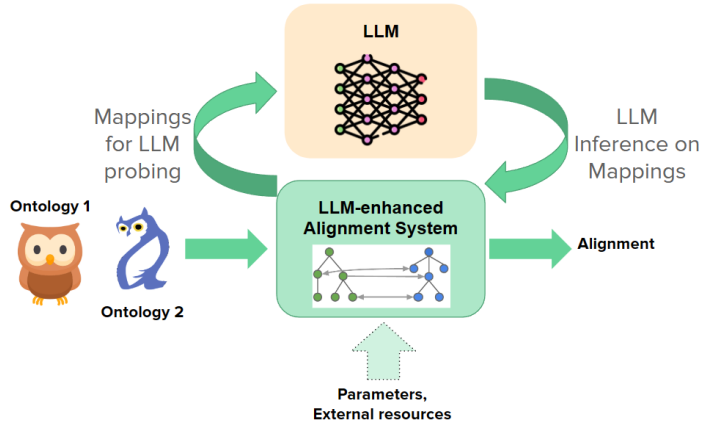
LLMs for Ontology Alignment (i)

- **OntoLAMA/probing** setting applied to inter-ontology subsumptions
- Key: successfully include **context in the prompts**.
- Potential templates:
  - *The source entity is C, the target entity is D. Are the concepts equivalent? <MASK>*
  - *The source entity is [a/an] C, a type of C', the target entity is [a/an] D, a type of D'. Are the concepts equivalent? <MASK>*

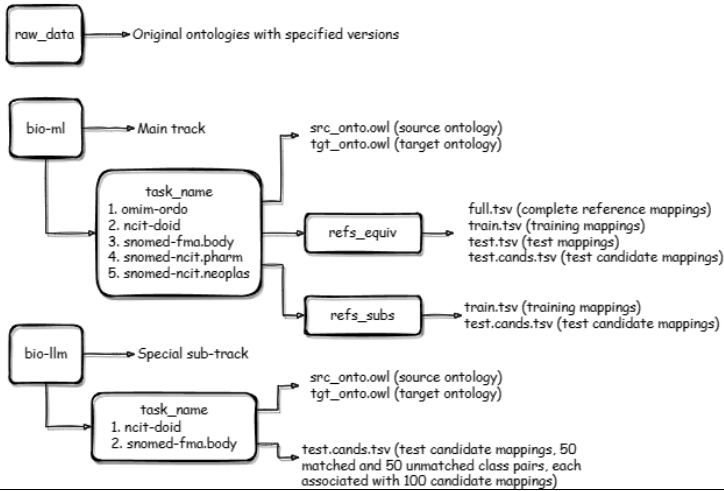
James Boyd. MSc Data Science @ City. Investigating OWL Ontology Alignment With Language Models.

LLMs for Ontology Alignment (ii)

LLM as Oracle or Domain Expert.



Benchmarking LLM-and-ML-Based OA Systems



Yuan He et al: Machine Learning-Friendly Biomedical Datasets for Equivalence and Subsumption Ontology Matching. ISWC 2022: 575-591

Acknowledgements

Acknowledgements

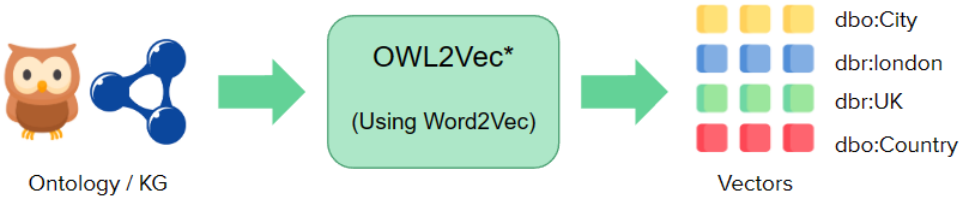
- DeepOnto developers:
  - **Yuan He** and **Ian Horrocks**, University of Oxford
  - **Jiaoyan Chen**, University of Manchester
  - **Hang Dong**, University of Exeter
- **James Boyd** (MSc Data Science)
- Referenced papers (images, ideas, etc.).
- Icons from `https://www.flaticon.com/free-icons/`

Laboratory Session

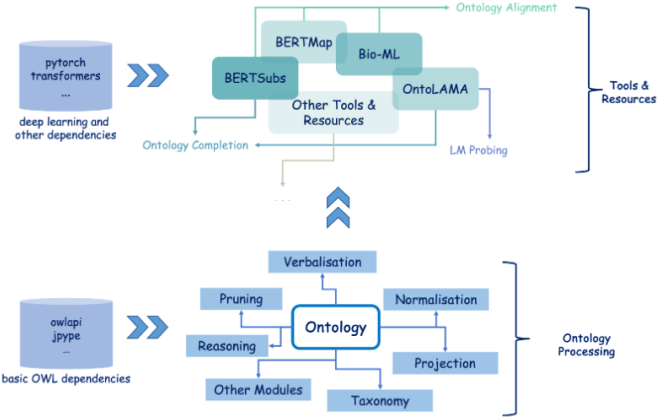
Lab Session

- Creating embeddings with OWL2Vec*.
  - Execute OWL2Vec* over the `Pizza` and `FoodOn` ontologies.
  - Compute similarity among words and entities.
  - Perform clustering and visualize results.
- Project:
  - A system for the OAEI (KG-to-KG).
  - A system for SemTab (CSV-to-KG).
- (Optional) Explore the DeepOnto library.

OWL2Vec* system



DeepOnto library



Yuan He, et al.: DeepOnto: A Python Package for Ontology Engineering with Deep Learning. Semantic Web Journal (2024)
`https://krr-oxford.github.io/DeepOnto/`

DeepOnto library: dependencies

- **OWL API** (Java-based) for basic ontology processing features.
- **PyTorch** for deep learning framework.
- **Huggingface Transformers** for language models.



Yuan He, et al.: DeepOnto: A Python Package for Ontology Engineering with Deep Learning. Semantic Web Journal (2024)
`https://krr-oxford.github.io/DeepOnto/`

Project submission

- Students need to work on a **project** (max 2 students per group). There are two options:
  - Create a (simple) system that performs KG to KG alignment.
  - Create a (simple) system that performs CSV to KG matching.
- Submission:
  - **When:** June 16, 23:59 CEST
  - **What:** a link to the GitHub repository where the system codes are
  - **How:** via a Google form (see GitHub)