From human experts to machines: An LLM supported approach to ontology and knowledge graph construction

by Vamsi Krishna Kommineni, Birgitta König-Ries & Sheeba Samuel

https://doi.org/10.48550/arXiv.2403.08345

Background: Ontologies and Knowledge Graphs

Ontologies = framework for describing and structuring domain knowledge, represent entities and relationships but also are the base for the construction of a KG

Knowledge Graphs = interlink information, provide reasoning and facilitate data analytics

Large Language Models (LLMs)

- LLMs are able to generate and interpret natural language
- massive pre-trained parameters and advanced neural architectures
- open-source LLMs offer transparency, model control, usage flexibility, and cost-effectiveness

What is the problem the researchers work on?

- ontologies & knowledge graphs rely on human domain expertise
- needs collaboration between multiple stakeholders (computer science, domain experts etc.)
- challenges: accuracy, scalability, and depth of knowledge captured
- difficult finding balance between resource-intensive construction and knowledge provided

What is their research question of the paper?

How can we (semi-)automatically construct a KG, from collecting competency questions to creating and populating an ontology, using LLMs, specifically for deep learning methodologies in biodiversity research?

What is their research question of the paper?

- do LLMs offer a possibility to automate (parts of) this process?
- focus on minimizing the time and human effort involved
- use of open-source LLMs for accessibility

Objectives and test case

Specific focus on:

- Automating competency question generation
- Ontology development
- Knowledge graph population
- Evaluation of results

with (semi-)automated construction through LLMs.

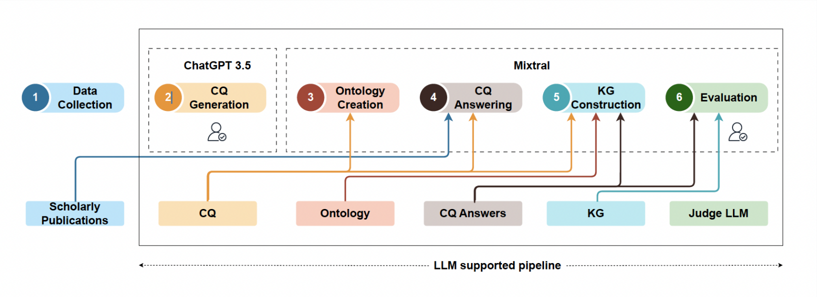
Test case: Deep Learning in the biodiversity domain.

What is a Competency Question?

= natural language questions outlining and constraining the scope of knowledge represented in an ontology; describe, what an ontology-based information system should be able to “answer”

Methodology: What the researchers did

1. Data Collection
2. Competency Question generation
3. Ontology creation
4. Competency Question answering
5. Knowledge Graph construction
6. Evaluation



Methodology: Data Collection

- reused a dataset of publications generated in their prior research
- systematic literature review to identify publications employing deep learning methods in biodiversity research
- 61 publication
- initial test with 5 publications

Methodology: Competency Question Generation

- Used ChatGPT-3.5 for initial competency question generation
- Human expert validation and refinement
- Resulted in 40 comprehensive questions

Methodology: Competency Questions generated by ChatGPT 3.5

automatic-KG-creation-with-LLM / CQs / CQs.txt

Vamsi-Kommineni Update CQs.txt 9a75299 · last month History

Code Blame 28 lines (28 loc) · 2.98 KB Raw

```
What methods are utilized for collecting raw data in the deep learning pipeline (e.g., surveys, sensors, public datasets)?
What data formats are used in the deep learning pipeline (e.g., image, audio, video, CSV)?
What are the data annotation techniques used in the deep learning pipeline (e.g., bounding box annotation, instance segmentation)?
What are the data augmentation techniques applied in the deep learning pipeline (e.g., Flipping, Rotating, Scaling)?
What are the datasets used in the deep learning pipeline (e.g., MNIST, CIFAR, ImageNet)?
What preprocessing steps are involved before training a deep learning model (e.g., normalization, scaling, cleaning)?
What are the criteria used to split the data for deep learning model training (e.g., train, test, validation)?
Where is the code repository of the deep learning pipeline available (e.g., GitHub, GitLab, BitBucket)?
Where is the data repository of the deep learning pipeline available (e.g., Zenodo, Figshare, Dryad, GBIF)?
What is the code repository link of the deep learning pipeline (e.g., Link to GitHub, GitLab, BitBucket)?
What is the data repository link of the deep learning pipeline (e.g., Link to Zenodo, Figshare, Dryad, GBIF)?
What type of deep learning model is used in the pipeline (e.g., CNN, RNN, Transformer)?
What are the hyperparameters used in the deep learning model (e.g., learning rate, optimizer)?
How are the hyperparameters of the model optimized (e.g., grid search, random search)?
What optimization techniques are applied in the deep learning pipeline (e.g., SGD, Adam)?
What criteria are used to determine when training is complete (e.g., validation loss plateau)?
What are the regularization methods used to prevent overfitting in the deep learning pipeline (e.g., dropout, L2 regularization)?
What is the strategy implemented to monitor the model performance during training?
Which frameworks are used to build the deep learning model (e.g., TensorFlow, PyTorch)?
Which hardware resources are used for training the deep learning model (e.g., GPUs, TPUs)?
What are the postprocessing steps involved after the model training (e.g., Saliency maps, Metrics calculation, Confusion matrix)?
What metrics are used to evaluate the performance of the deep learning model (e.g., accuracy, precision, recall)?
What measures were taken to ensure the generalizability of the deep learning model (e.g., Diverse dataset, cross-validation, Stratified splitting)?
What strategies are employed to handle randomness in the deep learning pipeline (e.g., random seed value)?
What is the purpose of the deep learning model (e.g., classification, segmentation, detection)?
What techniques are used to address data bias during preprocessing of the deep learning pipeline (e.g., Stratified splitting, oversampling, undersampling, Diverse data collection)?
What process was followed to deploy the trained deep learning model (e.g., Model serialization, Platform selection)?
Which platform was used to deploy the deep learning model (e.g., AWS, Azure, Google Cloud platform)?
```

Figure 2: 40 Competency Questions on Deep Learning in biodiversity by ChatGPT 3.5 (Source: GitHub)

Methodology: Ontology Creation

- manual comparison of the content quality of Llama 2-70B, Mixtral 8x7B9 & Falcon-40B
- ended up using Mixtral 8x7B
- two-step strategy

Methodology: Ontology Creation

1. extract all concepts and their relationships from the competency questions
2. constructed an ontology for describing information on deepl learning pipelines

- using PROV-O ontology as a foundational ontology

What is PROV-O?

- W3C standard ontology
- for representing and exchanging provenance information
- provides a set of classes, properties, and restrictions

Methodology: Competency Question answering

- Retrieval-Augmented Generation (RAG) approach
- retrieved answers for all the competency questions from the first 5 selected biodiversity publications
- Text processing to refine the generated answers
- elimination of redundant content

Methodology: Knowledge Graph Construction

- processed CQ answers + questions and the LLM-generated ontology were given as input to the LLM
- prompt instructed the LLM to extract key entities, relationships, and concepts from the answers
- created multiple KGs with two different prompts and two different RAG-generated CQ answers

Methodology: Evaluation

- used an LLM as a judge to score the generated outputs
- ground truth was created by a human expert for five scholarly articles (annotad CQ ground truth text + labeled CQ answers)
- LLM checked if the extracted KG concepts appeared in the respective generated CQ answers
- limited dataset of five scholarly articles to minimize time and manual annotation effort

What are their results?

Key achievements:

- generated 40 comprehensive CQs
- DLProv Ontology with 45 classes and 41 relationships
- 365 axioms in total

What are their results?

Evaluation metrics:

- 42 disagreements between human annotator and LLM judge out of 200 evaluated CQ answers
- across all KGs they successfully correlated 142 KG individuals out of 203 total
- variable performance across different publications (61.67% - 91.53%)

Challenges and Limitations

- LLM hallucination and prompt sensitivity
- consistency variations in generated KGs
- redundant entity creation
- Hardware dependency (variation in results)
- limited ontology reuse

Future Work

- testing with different hardware configurations
- evaluation with various open-source LLMs
- enhanced prompt engineering
- better mapping with existing ML/DL ontologies
- improved pipeline automation

Their Conclusion

- LLMs show promise as assistants in ontology and KG creation
- significant reduction in required human effort
- opens possibilities for broader adoption of semantic web technologies

Discussion from my side

- test case based on just five articles; limited generalization or validation of the approach
- using an LLM to evaluate outputs might introduce circular reasoning
- evaluation metrics focus on entity correlation and agreement rates but not on the semantic or functional quality
- LLM hallucination: paper acknowledges this issue but doesn't provide a detailed analysis of its impact
- concerns about reproducibility and consistency (LLMs / Hardware etc.)

Sources

1. From human experts to machines: An LLM supported approach to ontology and knowledge graph construction
2. PROV-O: The PROV Ontology
3. GitHub Repository: Automatic KG Creation with LLM