The Knowledge Graph for End-to-End Learning on Heterogeneous Knowledge

Xander Wilcke
w.x.wilcke@vu.nl

Peter Bloem
p.bloem@vu.nl

Victor de Boer
v.de.boer@vu.nl
Our Position

- Expressing *heterogeneous knowledge* using *knowledge graphs* allows us to build true end-to-end models across domains and tasks.

- We should adopt the *knowledge graph* as the *default* data model for machine learning on *heterogeneous knowledge*. 
A typical machine learning workflow
A typical machine learning workflow

Data → Preprocessing → Feature Engineering → Model Fitting

Feature engineering introduces bias
Towards End-to-End Learning

With end-to-end learning

- Every step in the pipeline is differentiable and can thus be tuned
- We can incorporate feature engineering into the model and it learn relevant features automatically
- We minimize bias otherwise introduced by the adding, removing, or transformation of data
However

- Current models only work well for a few specific domains; for many other domains we first need to create new models.

- Current models are unsuited for heterogeneous knowledge.
However

- Current models **only work well for a few specific domains**; for many other domains we first need to create new models.
- Current models are **unsuited for heterogeneous knowledge**.

Solution

a) adopt the knowledge graph as default data model for this kind of knowledge, and

b) develop end-to-end models which can directly consume knowledge graphs.
The Knowledge Graph

Encodes knowledge using binary statements of the form

\[(subject, predicate, object)\]

Example:

(Kate, knows, Mary)
(Kate, lives_in, Amsterdam)
(Mary, age, 32)
(Kate, status, “at work”)
Can be visualized as a graph:
Motivation for the knowledge graph as data model

- Naturally encodes heterogeneous knowledge

- A single uniform data model across nearly all domains and tasks

- Any method that is tailored to knowledge graphs can consume all knowledge graphs without preprocessing them first

- Data sets expressed as knowledge graphs are task independent
The Knowledge Graph as Data Model for Machine Learning

Also,

- Greatly simplifies the integration of datasets
- Provides a natural way to integrate different forms of background knowledge
A huge collection of knowledge graphs already exists, and is freely available on the Linked Open Data cloud.
End-to-End Learning on knowledge graphs is still very experimental and still has many unsolved challenges.

We identify four major challenges:

1) Dealing with *implicit* knowledge
2) Dealing with *incomplete* knowledge
3) Dealing with *differently-structured* knowledge
4) Dealing with *multi-modal* knowledge
Heterogeneous knowledge is multi-modal by nature

- Multi-modal learning on knowledge graphs has been left largely unaddressed
- We lose potentially-relevant information
Our approach:

- **Extend RGCN** [1] with modules dedicated to different modalities, each one dealt with accordingly and projected into the same multi-modal embedding space.