A Deep Neural Network for Link Prediction on Knowledge Graphs

Wilcke, W.X.

2016

A Deep Neural Network for Link Prediction on Knowledge Graphs

VU UNIVERSITY AMSTERDAM


Xander Wilcke
w.x.wilcke@vu.nl


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


Michel de Kleijn
mtm.de.kleijn@vu.nl


Frank van Harmelen
frank.van.harmelen@vu.nl

Aim of this Research

Recent years have seen the emergence of graph-based Knowledge Bases build upon **Semantic Web** technologies, known as **Knowledge Graphs**. Effectively learning from these complex relational structures remains a challenge yet to be overcome.

For this purpose, we are investigating the effectiveness of **Link Prediction** through means of **Deep Learning an Artificial Neural Network**, thereby using a **Self-Organizing Semantic Map** as input. To optimize both learning method and model we are exploring **Bayesian Hyper-parameter optimization**.

During evaluation, special attention will be given to the usefulness of made predictions to domain experts.

Knowledge Graphs

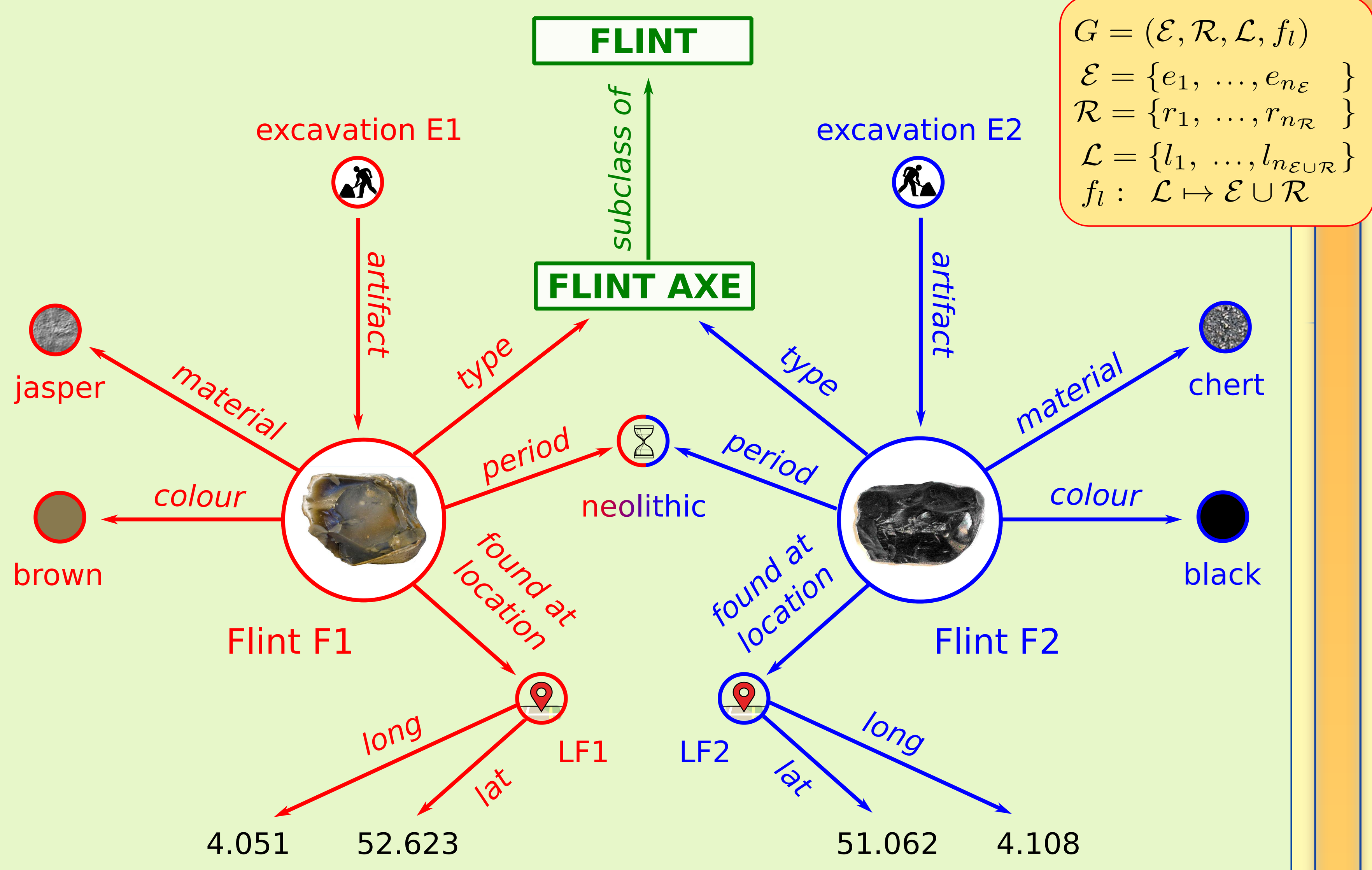
- Describe **factual information as relations between entities** (edges between vertices)
- Relations and entities are assigned **special labels which guard their semantics**
- Semantic labels are strictly defined in common ontologies (schemes)
- **Ambiguity is minimized** within and between Knowledge Graphs by freely sharing ontologies on the (Semantic) Web
- Inherit **Deductive Reasoning capabilities** from their underlying formal system

Example

Semantic Web

A Knowledge Graph contains a finite set of entities and a finite set of relations, as well as a finite set of labels and a corresponding mapping over the graph's elements.

$G = (\mathcal{E}, \mathcal{R}, \mathcal{L}, f_l)$
$\mathcal{E} = \{e_1, \ldots, e_{n_\mathcal{E}}\}$
$\mathcal{R} = \{r_1, \ldots, r_{n_\mathcal{R}}\}$
$\mathcal{L} = \{l_1, \ldots, l_{n_{\mathcal{E}\cup\mathcal{R}}}\}$
$f_l : \mathcal{L} \mapsto \mathcal{E} \cup \mathcal{R}$



A small Knowledge Graph depicting two distinct flints (red and blue) found during two separate archaeological excavations. Both flints were found to be of the *flint axe* class (rectangles), which is a subclass of the *flint* class. Each flint has various properties (cursive labels), which may take a value from a finite (circles) or infinite set (black text). Some relations might be left undefined, as their existence is simply unknown due to the **Open World Assumption**.

Queries fired over these data are able to answer questions like :

- List all flints of type *flint axe* which are of colour black and from period neolithic.
- List all brown flints found within 10 m from a black flint from the same period.
- List all excavations in the Netherlands during which chert *flint axes* were found.

Forthcoming Research

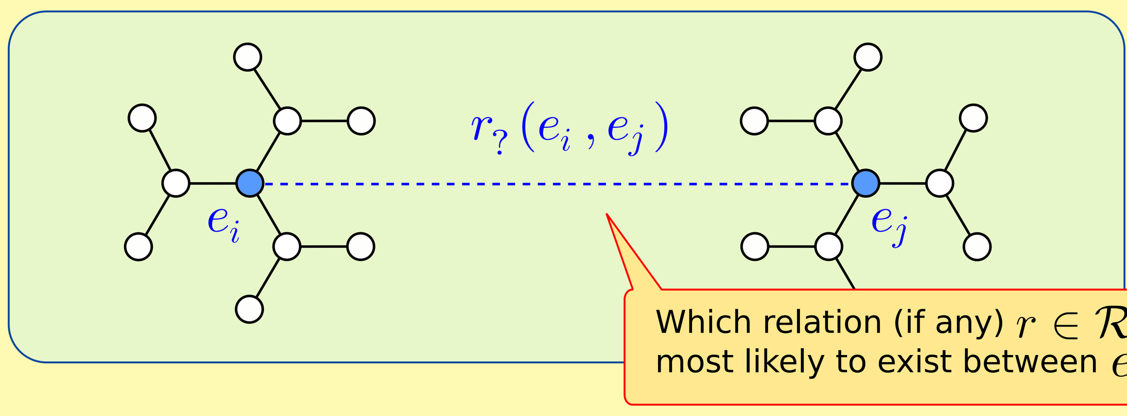
- **Minimizing the information loss** caused by propositionalization.
- **Investigate** the trade-off between generic and domain-specific **exploitation of ontological features**.
- Extensively **evaluate** our approach **on real-world Knowledge Graphs**.

Motivation

By default, reasoning over Knowledge Graphs is

- **completely dependent on axiomatic prior knowledge, and**
- **solely able to derive information that was already implicitly present.**

Hence, the existence of many relations (= facts) remains unknown



Which relation (if any) $r \in \mathcal{R}$ is most likely to exist between e_i and e_j ?

Using Link Prediction, we can estimate these relations' probability of existence

$$\widehat{r(e_i, e_j)} = f_m(e_i, e_j)$$

$$f_m(e_i, e_j) = \arg\max \{r \mid r \in \mathcal{R} \wedge \forall s \in \mathcal{R} : P(r(e_i, e_j)) \geq P(s(e_i, e_j))\}$$

Example of single-class prediction using proposed model f_m

Knowing which relations are likely to exist is **highly welcomed by Domain Experts** :

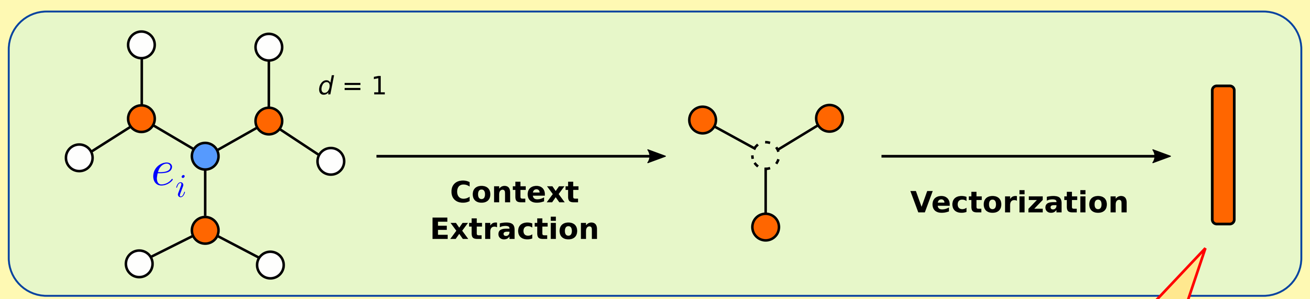
- as starting points from which they form new potential hypotheses
- as potential evidence to support proving of disproving existing hypotheses
- as measure to evaluate the trustworthiness of potential sources of information

Methodology

Propositionalization Strategy

Generated contextual vectors hold information on :

- the **labels** of the sample's elements (embedded features)
- the **semantics** of the sample's elements (ontological features)
- the **local neighbourhood** up to depth *d* of the sample's entities (graph features)

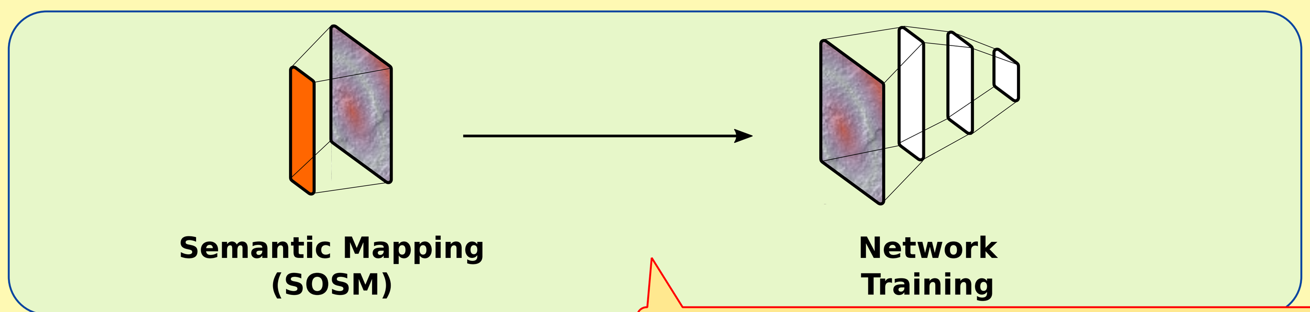


Semantic Context Vector

Learning and Optimization

Training the model involves :

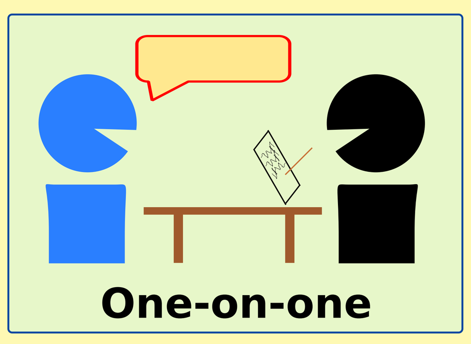
- constructing a **Self-Organizing Semantic Map** (SOSM) from the **contextual input vectors** to serve as input to the network
- **pre-training** the weight matrices using **Restricted Boltzmann Machines**, followed by **finetuning** them using **supervised Back Propagation**
- **Bayesian Hyper-parameter Optimization** using **Random Forests** as prior

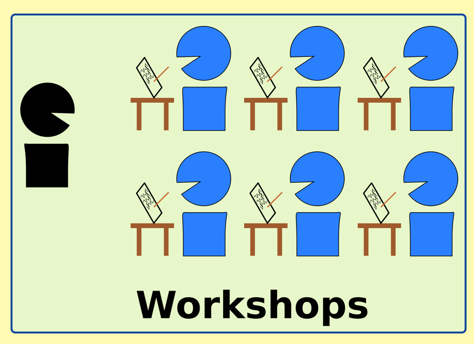


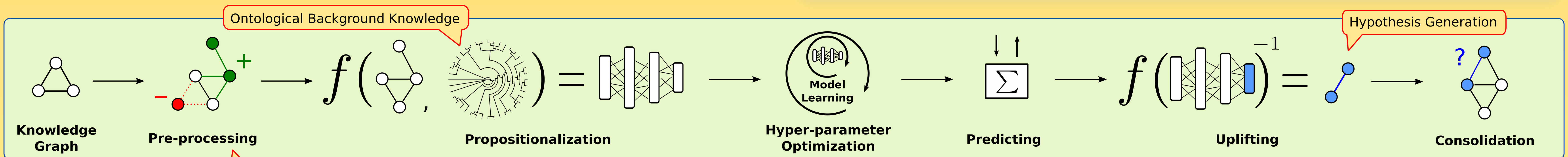
1) learn a SOSM from the contexts of all $e \in \mathcal{E}$
2) train a deep neural net using the SOSM as input

Evaluating Predictions

- area under the precision-recall curve (**AUC-PR**)
- **usefulness of predictions to domain experts** (qualitative measures)







NETWORK INSTITUTE

SPIN LAB SPATIAL INFORMATION LABORATORY

Universiteit Leiden

Acknowledgements: This research is partially funded by the ARIADNE project through the European Commission under the Community's Seventh Framework Programme, contract no. FP7-INFRASTRUCTURES-2012-1-313193.

SEVENTH FRAMEWORK PROGRAMME

ARIADNE
www.ariadne-infrastructure.eu