Deductive Reasoning for Cross-Knowledge Graph Entailment

Ebrahimi M, Sarker MK, Bianchi F, Xie N, Eberhart A, Doran D, Kim H, Hitzler P

by
Aniruddha Datta
Ethan Clark
Viswa Ivaturi

Deductive Reasoning

Description and Types of Inference

- Deductive reasoning is simply reaching conclusions based on premises that are known or assumed to be true.
- Premise A : All dogs are mammals

  Premise B : Mammals are warm blooded

  Conclusion: All dogs are warm blooded
- Usually happens by applying “rules of inference” or schema which are provided mostly by classical logic
- Examples of well-known rules of inference are as follows
- *Modus ponens*: if $P \to Q, P \vdash Q$ in sequent notation and $((P \to Q) \land P) \to P$ in propositional logic notation

  where, P ,Q and $P \to Q$ are statements (or propositions) in a formal language and $\vdash$ is a metalogical symbol meaning that Q

  is a syntactic consequence of and $P \to Q$ in some logical system.
- *Modus Tollens: if* $P \to Q, \neg Q \vdash \neg P$ in sequent notation and $((P \to Q) \land \neg Q) \to \neg P$

  in propositional logic notation

Description and Types of Inference

- *Hypothetical syllogism : if* $\frac{P \vdash Q \quad Q \vdash R}{P \vdash R}$ in sequent notation and in propositional logic :

$$((P \to Q) \land (Q \to R)) \to (P \to R)$$

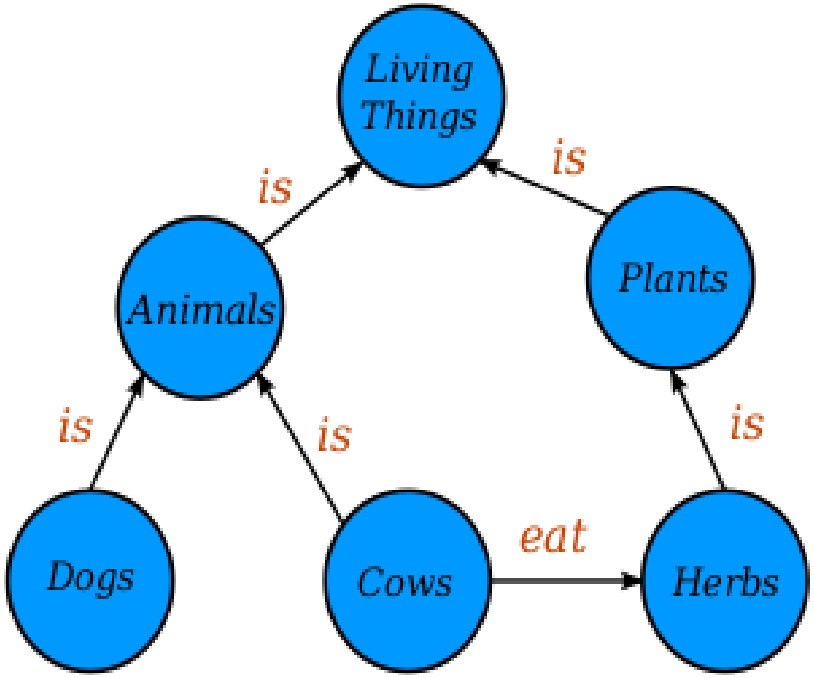
- Validity and Soundness

- An argument is valid if it is impossible for the premises to be false if the conclusion is true.
- An argument is sound if it is valid and its premises are true.

Knowledge Graphs

Description and Uses

- A knowledge graph is a directed labelled graph that has domain specific meanings attached to it.



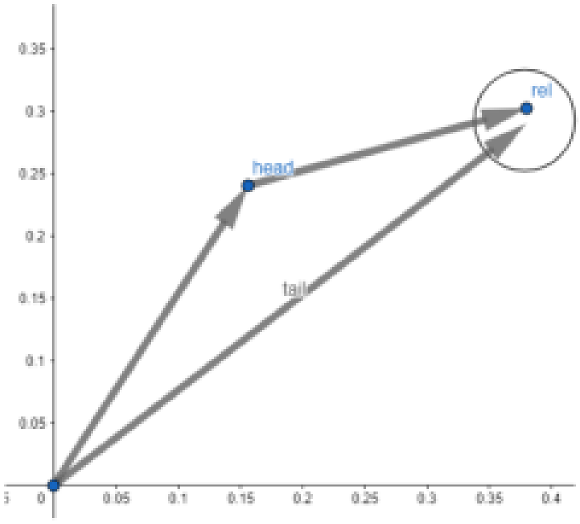
Description and Uses

- Used to structure knowledge and manipulate massive amounts of data using graph algorithms.
- Enterprises and social media use it to maintain customer relations
- In the context of AI, it is also called Semantic Networks, KGs are used for a wide range of tasks like knowledge representation, classification and Reasoning.
- World Wide Web Consortium (W3C) has standardized a family of knowledge representation languages
- These languages include the Resource Description Framework(RDF), the Web Ontology Language(OWL), and the Semantic Web Rule Language (SWRL)
- The authors have used RDF for their experiments

Knowledge Graph Embeddings

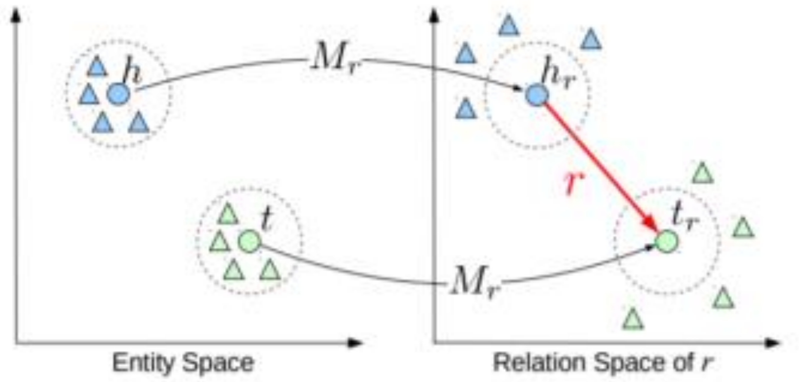
Methods of Embedding

- Syntactic Normalization which renames constants, variables, predicates, etc. to predefined syntactical names across all domains of normalisation. Used by the authors
- TransE : If the subject, object and relation holds, we can describe objectas as a vector addition of subject and relation assuming they are in the same vector space, $\mathbb{R}^k$.



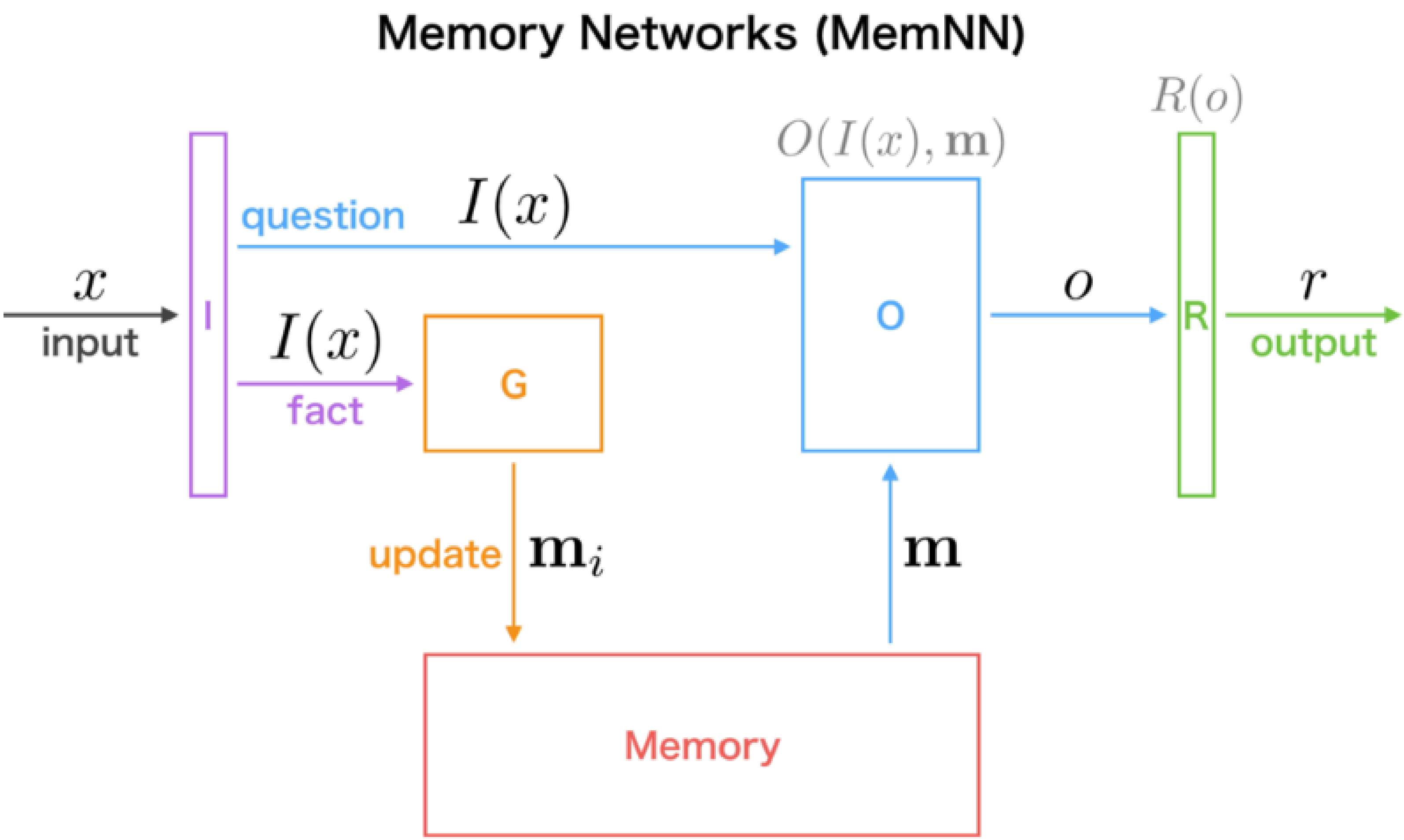
Methods of Embedding

- TransR : Does not assume subject and relation to be in same vector space and thus having an entity space, $(h, t) \in \mathbb{R}^{k}$ and relations having a relation space $\mathbb{R}^{d}$ where $d \neq k$ and we operate in the projection matrix of the two spaces $\boldsymbol{M_r} \in \mathbb{R}^{k \times d}$.



Memory Networks

Structure of Memory Networks



Structure of Memory Networks

- It contains an indexed memory(m) and 4 modules which are namely
- I (input feature maps): Converts the incoming input to internal feature maps.
- G(generalization): Updates the old memory give a new set of inputs
- O(output feature map): produces a new output (in the feature representation space), given the new input and the current memory state.
- R(response): converts the output into the response format desired. For example, a textual response or an action

Inference of Memory Networks

- Core of inference is the Output features module and response. Where a scoring function is used to find the best supporting memory.

$$o_1 = O_1(x, \mathbf{m}) = \underset{i=1,\dots,N}{\arg\max}\; s_O(x, \mathbf{m}_i)$$

- For the next iteration we use the same process but with an array of x and m1

$$o_2 = O_2(x, \mathbf{m}) = \underset{i=1,\dots,N}{\arg\max}\; s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_i)$$

- Simplest example of a response model is given by

$$r = \operatorname{argmax}_{w \in W}\; s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], w)$$

- And the scoring function can be given by embedding functions

$$s(x, y) = \Phi_x(x)^\top U^\top U \Phi_y(y).$$

Training/Learning in Memory Networks

- For text inputs, training was done using the Margin ranking loss and Stochastic Gradient Descent.
- Minimizing the loss over the parameters Uo and Ur.

$$\sum_{\bar{f} \neq \mathbf{m}_{o_1}} \max(0, \gamma - s_O(x, \mathbf{m}_{o_1}) + s_O(x, \bar{f})) +$$

$$\sum_{\bar{f}' \neq \mathbf{m}_{o_2}} \max(0, \gamma - s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_{o_2}]) + s_O([x, \mathbf{m}_{o_1}], \bar{f}'])) +$$

$$\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], r) + s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \bar{r}]))$$

Logic Entailment

Problem Formulation

- Any logic $\mathcal{L}$ comes with an entailment relation

$$\models_{\mathcal{L}} \subseteq T_{\mathcal{L}} \times F_{\mathcal{L}}$$

where $F_{\mathcal{L}}$ is a subset of set of all logical formulas (or axioms) over $\mathcal{L}$, and $T_{\mathcal{L}}$ is the set of all theories over $\mathcal{L}$.

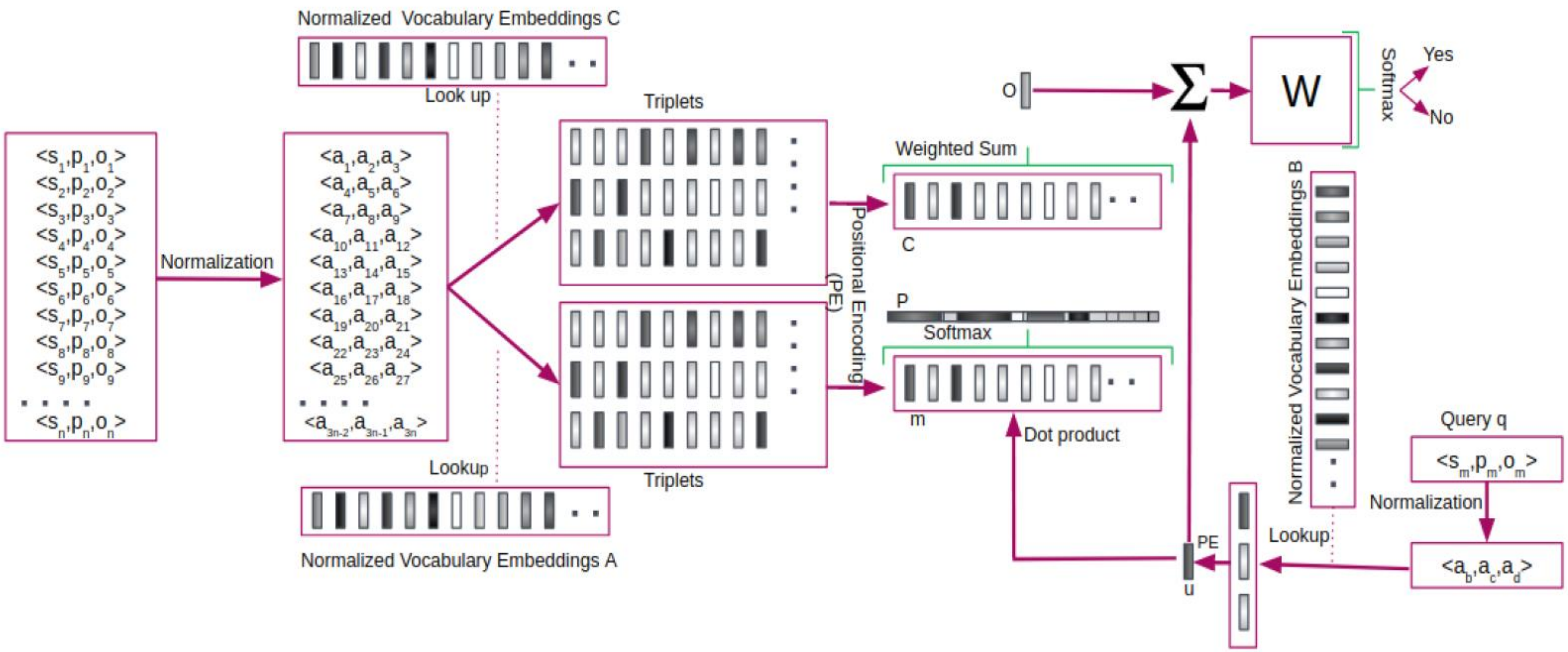
- T entails F or F is entailed by T if $T \models F_{\mathcal{L}}$
- Reframing the deductive reasoning problem as a classification task
- Train a model on a set of theories (T, F) to see if $(T, F) \in T_{\mathcal{L}} \times F_{\mathcal{L}}$ is a valid entailment
- If successful, transfer that learning to new theories over the same logic
- If not then $T \not\models_{\mathcal{L}} F$

Model Summary

Model Architecture

- Takes a discrete set *G* of normalized RDF triplets $t_1,\ldots t_n$ that are stored in memory, a query *q,* and outputs a “yes” or “no” answer

- normalized t_i and *q* contains symbols from general dictionary with *V* normalized words shared among all normalized RDF theories



Model Description

- Model stores the embeddings of the normalized triplets in our KG in an external memory component
- This component is defined as *n x d* tensor where

  n = number of triplets

  d = dimensionality of the embeddings
- Memory vector stores 2 continuous representations of m_i and c_i obtained from matrices A and C with size *d x V*, V = size of vocabulary
- Query *q* is embedded via a matric *B* to obtain internal state *u*

Model Description

- Attention mechanism for *q* over memory input is represented by a SoftMax

$$p_i = \text{Softmax}(u^T(m_i))$$

$$\text{where} \qquad \text{Softmax}(a_i) = \frac{e^{(a_i)}}{\sum_j e^{(a_j)}}$$

- The above equation calculates probability vector *p* over the memory input
- The output vector *o* is a weighted sum of memory content ci with respect to corresponding probabilities pi

$$o = \sum_i p_i c_i$$

- The internal state of the query vector updates for next hop as $u^{k+1} = u^k + o^k$, and the process repeats K times

Syntactic Normalization

- Deductive reasoning requires the names of entities to be insubstantial
- Embeddings that are agnostic to the strings used as primitives in the KG are built
- Syntactic normalization – renaming of primitives such as variables, constants, functions, predicates to a predefined entity names
- By random renaming, the network will learn the structure within the theory and not the actual names
- Helps in forgetting irrelevant label names
- Assists in transfer learning from one KG to the other

Cross KG entailment

- Resource Description Framework (RDF) is a widely used Web standard for data publishing and expressing KGs
- RDF KG stores any statement as triplets (e1,r,e2)

  e1 - Subject

  e2 – Object

  r – relation binding e1 and e2
- Syntactic normalization of the elements in triplets uses Position Encoding (PE)
- PE encodes position of each element within a triplet

Cross KG entailment

- j^{th} element of i^{th} triplet be t_{ij}
- Memory vector representation of each triplet is

$$m_i = \sum_j l_j \circ t_{i,j}$$

$$l_{k,j} = (1 - j/3) - (k(1 - 2j/3)/d)$$

k = number of hops

d = size of embeddings

3 = number of RDF elements

- Hence each memory slot is position-weighted summation of each triplet
- PE ensures the order of elements affects the encoding of each memory slot

Evaluation

Dataset

- Data collected from RDF datasets from Linked Data cloud and Data Hub
- Training dataset - RDF KGs each of size 1000 triplets sampled from 20 Web Ontology Language(OWL)
- Test dataset - Linked Data test set and custom created small dataset with long reasoning chains
- For each KG a finite set of inferred triplets were created with some positive and invalid class instances
- Invalid class method 1 – random permutation and removing entailed triplets
- Invalid class method 2 called a – one random element in each valid triplet is replaced with another random element based on position encoding

Training and Evaluation

- Training batch size of 100 triplets
- Final batch queries get zero-padded to reach maximum batch size of 100
- Dimensionality of embedding = 20
- Hence the matrices of A,B, and C are of size |V| x 20
- K = 10
- Evaluation metrics are average of precision, recall and f-measure over all the KGs
- Includes valid and invalid triplets with true negatives recall specifically counted

Evaluation

- Non-normalized embedding version memory network is considered as baseline

Training Dataset	Test Dataset	Valid Triples Class			Invalid Triples Class			Accuracy
		Precision	Recall /Sensitivity	F-measure	Precision	Recall /Specificity	F-measure	
OWL-Centric Dataset	Linked Data	93	98	96	98	93	95	**96**
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	88	91	89	90	88	89	**90**
OWL-Centric Dataset	OWL-Centric Test Set [b]	79	62	68	70	84	76	**69**
OWL-Centric Dataset	Synthetic Data	65	49	40	52	54	42	**52**
OWL-Centric Dataset	Linked Data [a]	54	98	70	91	16	27	86
OWL-Centric Dataset [a]	Linked Data [a]	62	72	67	67	56	61	91
OWL-Centric Dataset(90%) [a]	OWL-Centric Dataset(10%) [a]	79	72	75	74	81	77	80
OWL-Centric Dataset	OWL-Centric Test Set [ab]	58	68	62	62	50	54	58
OWL-Centric Dataset [a]	OWL-Centric Test Set [ab]	77	57	65	66	82	73	73
OWL-Centric Dataset	Synthetic Data [a]	70	51	40	47	52	38	51
OWL-Centric Dataset [a]	Synthetic Data [a]	67	23	25	52	80	62	50
Baseline								
OWL-Centric Dataset	Linked Data	73	98	83	94	46	61	43
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	84	83	84	84	84	84	82
OWL-Centric Dataset	OWL-Centric Test Set [b]	62	84	70	80	40	48	61
OWL-Centric Dataset	Synthetic Data	35	41	32	48	55	45	48

Results

- Big difference in training times for embedded matrices in the original and normalized cases
- Example – Embedded matrices for normalized OWL-centric training dataset is 3033x20
- For non-normalized one is it 811,261x20
- Normalization reduces the space required for saving by 80 times($\approx$ 4GB)
- It is also 40 times faster to train than non-normalized ones
- Normalized model trained for a day and achieved better accuracy
- The non-normalized model took a week to train on a non-normalized dataset but still achieved less accuracy

Shortcoming and Future Work

- Poor performance when tested against a tricky version of Linked Data
- Tricky version has negative instances bear close resemblance with positive ones
- The model is not as good against a challenging synthetic data
- It is due to the difference in length distribution of original training data and the synthetic data reasoning hops
- It is concluded also from the previous studies that the reasoning chain length in real-world KGs is limited to 3 or 4
- For future work, the authors plan to investigate the model scalability and its adaptability to complex, synthetic datasets

References

- Ebrahimi M, Sarker MK, Bianchi F, Xie N, Eberhart A, Doran D, Kim H, Hitzler P. Neuro-Symbolic Deductive Reasoning for Cross-Knowledge Graph Entailment. InAAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering 2021 Mar 22.
- B. Makni, J. Hendler, Deep Learning for Noise-tolerant RDFS Reasoning, Ph.D. thesis, Rensselaer Polytechnic Institute, 2018.
- S. Sukhbaatar, J.Weston, R. Fergus, et al., End-to-end memory networks, in: Advances in neural information processing systems, 2015, pp. 2440–2448.
- W. Xiong, T. Hoang,W. Y.Wang, Deeppath: A reinforcement learning method for knowledge graph reasoning, arXiv preprint arXiv:1707.06690 (2017).

Ira A. Fulton Schools of Engineering

Arizona State University