Deductive Reasoning for Cross-Knowledge Graph Entailment


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 \( P \rightarrow Q, P \vdash Q \) in sequent notation and \( ((P \rightarrow Q) \land P) \rightarrow P \) in propositional logic notation

  where, \( P, Q \) and \( P \rightarrow Q \) are statements (or propositions) in a formal language and \( \vdash \) is a metalogical symbol meaning that \( Q \) is a syntactic consequence of \( P \rightarrow Q \) in some logical system.

• **Modus Tollens**: if \( P \rightarrow Q, \neg Q \vdash \neg P \) in sequent notation and \( ((P \rightarrow Q) \land \neg Q) \rightarrow \neg P \) in propositional logic notation
Description and Types of Inference

• *Hypothetical syllogism*: if \( P \vdash Q \) \( Q \vdash R \) in sequent notation and in propositional logic:

\[
\frac{P \vdash R}{((P \rightarrow Q) \land (Q \rightarrow R)) \rightarrow (P \rightarrow R)}
\]

• 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 object 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 \(M_r \in \mathbb{R}^{k \times d}\).
Memory Networks
Structure of Memory Networks

Memory Networks (MemNN)

$I(x)$

$I(x)$

update $m_i$

$m$

Memory

$r$

output

$O(I(x), m)$

$O$
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 given 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.

\[ o_1 = O_1(x, m) = \arg \max_{i=1,...,N} s_O(x, m_i) \]

• For the next iteration we use the same process but with an array of \( x \) and \( m_1 \)

\[ o_2 = O_2(x, m) = \arg \max_{i=1,...,N} s_O([x, m_{o1}], m_i) \]

• Simplest example of a response model is given by

\[ r = \arg \max_{w \in W} s_R([x, m_{o1}, m_{o2}], w) \]

• 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 $U_o$ and $Ur$.

\[
\sum_{\bar{f} \neq m_{o_1}} \max(0, \gamma - s_O(x, m_{o_1}) + s_O(x, \bar{f})) + \\
\sum_{\bar{f}' \neq m_{o_2}} \max(0, \gamma - s_O([x, m_{o_1}], m_{o_2}) + s_O([x, m_{o_1}], \bar{f}'])) + \\
\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, m_{o_1}, m_{o_2}], r) + s_R([x, m_{o_1}, m_{o_2}], \bar{r}']))
\]
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 $n \times d$ 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 $d \times V$, $V =$ size of vocabulary

• Query $q$ is embedded via a matrix $B$ to obtain internal state $u$
Model Description

• Attention mechanism for $q$ over memory input is represented by a SoftMax

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

where

\[ \text{Softmax}(a_i) = \frac{e^{(a_i)}}{\sum_j e^{(a_j)}} \]

• The above equation calculates probability vector $p$ over the memory input

• The output vector $o$ is a weighted sum of memory content $c_i$ with respect to corresponding probabilities $p_i$

\[ o = \sum_i p_i c_i \]

• 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

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

\[
m_i = \sum_j l_j \cdot t_{i,j}
\]

\[
l_{k,j} = (1 - j/3) - (k(1 - 2j/3)/d)
\]

\( k = \) 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 $|V| \times 20$
• $K = 10$
• 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

<table>
<thead>
<tr>
<th>Training Dataset</th>
<th>Test Dataset</th>
<th>Valid Triples Class</th>
<th>Invalid Triples Class</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall /Sensitivity</td>
<td>F-measure</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Linked Data</td>
<td>93</td>
<td>98</td>
<td>96</td>
</tr>
<tr>
<td>OWL-Centric Dataset (90%)</td>
<td>OWL-Centric Dataset (10%)</td>
<td>88</td>
<td>91</td>
<td>89</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>OWL-Centric Test Set b</td>
<td>79</td>
<td>62</td>
<td>68</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Synthetic Data</td>
<td>65</td>
<td>49</td>
<td>40</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Linked Data a</td>
<td>54</td>
<td>98</td>
<td>70</td>
</tr>
<tr>
<td>OWL-Centric Dataset a</td>
<td>Linked Data a</td>
<td>62</td>
<td>72</td>
<td>67</td>
</tr>
<tr>
<td>OWL-Centric Dataset (90%)</td>
<td>OWL-Centric Dataset (10%) a</td>
<td>79</td>
<td>72</td>
<td>75</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>OWL-Centric Test Set ab</td>
<td>58</td>
<td>68</td>
<td>62</td>
</tr>
<tr>
<td>OWL-Centric Dataset a</td>
<td>OWL-Centric Test Set ab</td>
<td>77</td>
<td>57</td>
<td>65</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Synthetic Data a</td>
<td>70</td>
<td>51</td>
<td>40</td>
</tr>
<tr>
<td>OWL-Centric Dataset a</td>
<td>Synthetic Data a</td>
<td>67</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Linked Data</td>
<td>73</td>
<td>98</td>
<td>83</td>
</tr>
<tr>
<td>OWL-Centric Dataset (90%)</td>
<td>OWL-Centric Dataset (10%)</td>
<td>84</td>
<td>83</td>
<td>84</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>OWL-Centric Test Set b</td>
<td>62</td>
<td>84</td>
<td>70</td>
</tr>
<tr>
<td>OWL-Centric Dataset</td>
<td>Synthetic Data</td>
<td>35</td>
<td>41</td>
<td>32</td>
</tr>
</tbody>
</table>
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 (≈ 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


