Relations, Data and Knowledge

Introduction to Knowledge graphs

The talk

- What are relations? in the context of NLP
- Data we find today
- What is Knowledge?
- Importance of Knowledge
- Knowledge Graphs and their representations
- Machine Learning work in the field of Knowledge Graphs
- Conclusion

(In each topic I will be talking a little about their respective research aspects)

What are relations?

In the context of NLP, a relation can be defined as a triple (s, p, o) where

S - Subject

O - Object

P - Predicate

Ex (IITGn, hosts, Blithchron), (Sudhir K Jain, directorOf, IITGn)

How is this related to NLP?

Multiple projects and tasks surrounding this concept. Will cover them as and when we come across those

However one of them is Relation Extraction?

Relation Extraction *- Extracting (s, p, o) triples given any text*

Ex - Given the sentence Prof Sudhir K jain is the director of IITGn - get

(Sudhir, directorOf, IITGn)

Simple?

Work which has been done in this field

- Machine Learning methods
  - **UnSupervised**
    - Using patterns
      - Hearst patterns (named after a prof in UCB)
        - https://www.aclweb.org/anthology/P18-2057/
        - http://people.ischool.berkeley.edu/~hearst/papers/coling92.pdf
      - Parse Trees
        - https://www.researchgate.net/publication/228905420_Triplet_extraction_from_sentences
      - Dependency methods
        - https://nlp.stanford.edu/software/openie.html
    - Open Domain Extraction

- **Semi Supervised**
  - Using Seed triples, analysing patterns and then finding other patterns (bootstrapping)
  - Using supervised techniques with Bootstrapping
    - GAN frameworks
- **Supervised**
  - Must work with a Dataset and a fixed set of relations
  - Unified Medical Language System defines
    - 134 different types of entities and 54 types of relations (becomes a classification problem)
    - Relation classification task
  - SemEval Task - sometimes has relation classification tasks
  - Automated Content Extraction defined 17 types of relations (2008)

http://aritter.github.io/courses/5525_slides/relation_extraction.pdf

Data we find -

- Structured
  - Tables
  - JSONs (Tree like structure)
- Unstructured (mostly)

Advantages of Tables?

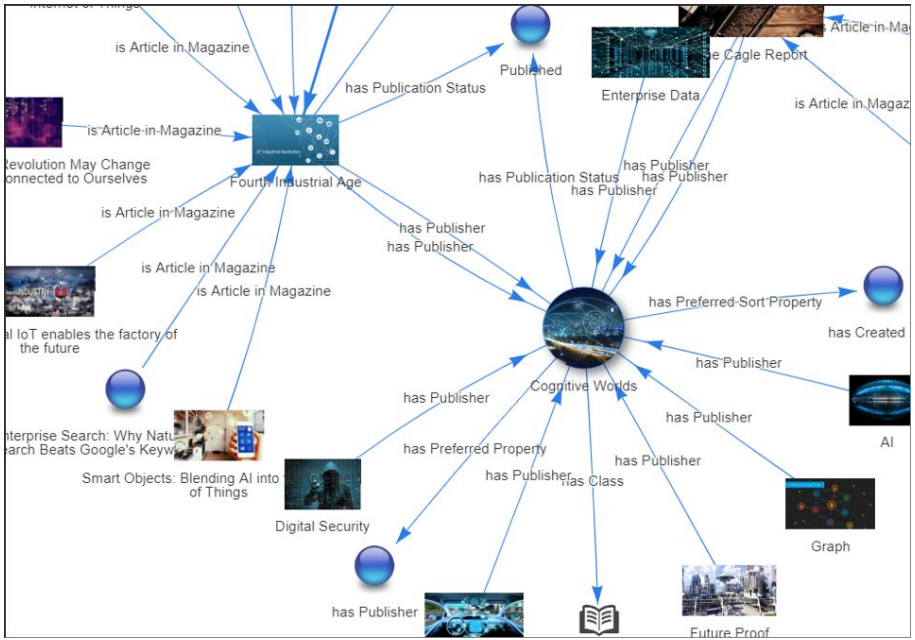
- Easy to use, modify, store

Disadvantages of Tables?

- Can’t directly infer without processing it

So what's Knowledge?

Knowledge is a form of structured data where we can directly infer between entities



Is this better?

Not completely

- Difficult to store and easy to interpret or the vice versa
- Softwares which would store them would be a little more complex
- Is of course not good for ML models which requires data in the form of Matrices

So why would a person/company use Knowledge instead of Data?

- More Information
- Easier interpretations
- Can perform simple yet powerful queries once preprocessed

Would making an already existing Dataset into Knowledge be beneficial?

Dataset

Images pixels(a0, a1, …), a-label

Knowledge Graph

a0 -Pixel1-> a-label (clusters)

a1 -Pixel2-> a-label

Will there be edges between clusters?

Better Q - Could you have edges between clusters?

Knowledge Graphs

- Graphs used to represent Knowledge (in a way similar to Graph databases, ie Graph databases can be modified to model Knowledge)

- In other words a set of interlinked relations

- If we are to preprocess this data and make it useful, we need representations right?

Revisiting Word Embeddings

- Why do we need word embeddings?

- Comes with a lot of other features such as contextual meanings, similarities, sentiment.
- So what if we are to represent Knowledge Graphs as embeddings, what can we do with this information?

Knowledge Graph Embeddings

sv - subject embedding vector

pv - predicate embedding vector

ov - object embedding vector

Imagine an operation f(sv, pv, ov) = y, where y can determine the confidence of a triple.

Different Methods/Losses

TransE

- $h + r \approx t$ when (h, r, t) holds
- score function:

$$f_r(h,t) = \|h + r - t\|_2^2$$

TransH

- $h, t \in R^k \quad r \in R^d \quad k \neq d$
- $M_r \in R^{k \times d}$
- Project entities from entity space to relation space:

$$h_r = hM_r \quad t_r = tM_r$$

TransR

- Score function:

$$f_r(h,t) = \|h_r + r - t_r\|_2^2$$

https://github.com/thunlp/KRLPapers

Use cases

Direct Use cases

- Ofcourse - checking validity of a triple
- Contextual meanings
  - Ex (dad, fatherOf, son) (dad, fatherOf, daughter) -> son and daughter likely to have same embeddings
- Finding new relations
  - Inverse of the above example
  - This can be done given you already have a Knowledge Graph, or Knowledge based vectors

Use Cases

Indirect Use cases

- Most important -> Improving Search!
- Robust Q&A frameworks

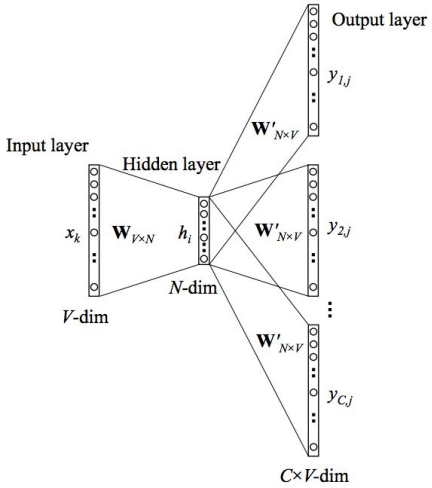
Applying Deep Learning to KGs - GNN

Motivation for Graph Neural Nets

- Helps capture Nodes and edges in a graph by creating an embedding for each of them (node embedding and edge embedding)
- Thus embeddings now contain information more than similarity, carry information about the neighbourhood.

DeepWalk (Node Embeddings)

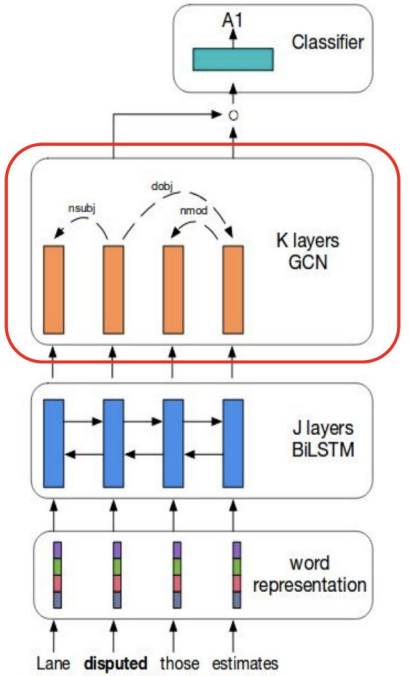
- Inspired by the skip-gram model in NLP
- Perform random walks on the graph in order to get node orders, (treat them like a sentence)
- Then use these node walks to find out context nodes and train a skip gram model on them.



Graph Neural Networks

GNN formulation by [Kipf et al., ICLR 2016]

$$h_v = f\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} W x_u + b\right), \quad \forall v \in \mathcal{V}.$$



Model with GCN as part of the network

NLP Tasks w.r.t. KGs

- KG population (finding missing items in a KG given extra information)
- KG completion (finding missing links)
- KG embeddings (Node and link embeddings)
- Open Domain Extraction