Entity Alignment

Group 1
Ann Tan, Steffen Brandenburg, Mustafa Yasin

Motivation

"AI systems make decision by using rules and logic to reason about **facts** or deduce new facts."

Knowledge Bases

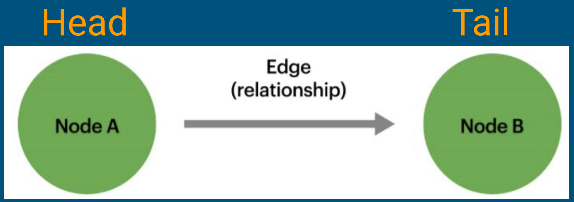
- Collection of facts/ground truths about the world.
- Structured information stored by a computer system.
- Used by expert systems for inference and deduction.

Internet as a knowledge base

- multiple sources in different languages
- i.e. Wikipedia, DBPedia, Wikidata, IMDB
- HOW TO to unify information? ➪ **Entity Alignment**

Knowledge Graph

- knowledge base represented as a graph
    - Entity - represented by itself
    - Relationship - inferred; with other entities, facts, circumstances
- a form of semantic network, limited to a specific domain.
- organized, easy to understand, to extract and to infer information from.



Source; https://medium.com/analytics-vidhya/introduction-to-knowledge-graphs-and-their-applications-fb5b12da2a8b

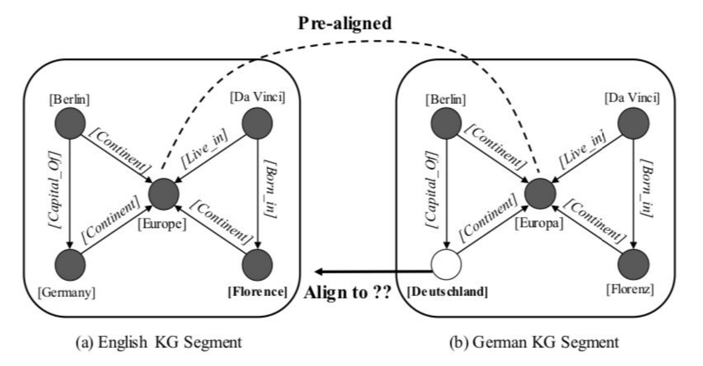
Uses of KGs:

1. Dialog Systems
2. Natural Language Generation
3. Question-Answering System
4. NER in Computational Argumentation

Entity Alignment

TASK: Find entities in 2 KGs that represent the same real-world entities.

1. <u>**Input**</u> includes:
    a. Two (2) KGs - **left** and **right**, each with:
        - List of <u>**entities**</u>, <u>**relations**</u>; some include attributes
        - **Triples** in the form (**head** entity, **relation**, **tail** entity)
    b. **Pre-aligned entities** from left KG to right KG



Source: https://dl.acm.org/doi/pdf/10.1145/3336191.3371804

2. Do <u>**supervised training**</u> using:
    a. Representation of entities / relations / attributes (i.e. **embeddings**).
    b. Triples to represent the **graph structure** (Adjacency, Degree).
    c. **Alignment** as supervision.
3. Predict correspondence of entities from both KGs using **similarity** metrics on the entities' representations.

Group Tasks

1. Analyze datasets.
2. Compare various approaches to KG Alignment.
   a. Read papers.
   b. Run published codes.
   c. Retrieve baseline results
3. Adapt from DBS Framework or implement new modules for the approaches in #2.
4. Test, Train and Evaluate.

Datasets

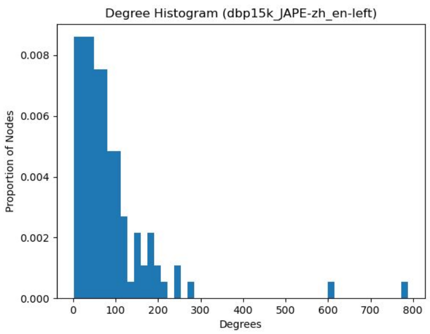
Sources	Name	Subset	Triple Size	Top Entities	Top Relations
Wikipedia (n-n)	WK3l15k WK3l120k	en-de en-fr	WK3l120k (en) largest with 1.3M triples	Countries Music Genre Sports Position	(25% - 47% of triples) Name Title Genre Birthplace
DBPedia (1-1)	DBP15k (Full) DBP15k (JAPE)	zh-en ja-en fr-en	DBP15k JAPE (zh) smallest with 70k triples	Countries Producer Record Label	(14% - 25% or triples) Starring Birthplace Writer / Producer
Wikipedia + Wikidata + DBPedia (1-1)	DWY100k	en-wd en-yg	Between 400-500k triples	Countries Year Sports Position	(36% - 76% of triples) Birthplace Year / Place of Death Name Team / Goals

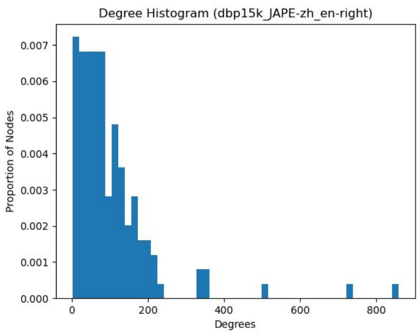
Datasets: Degree

<u>DBP15k (JAPE) zh-en</u>

- Avg Degree: 6.57 | 8.57

- A very small proportion of nodes have > 600

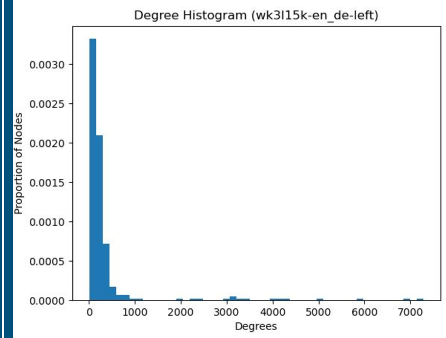


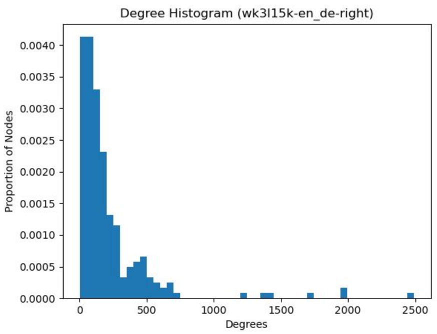


<u>WK3l15k en-de</u>

- Avg Degree: 21.62 | 16.51

- Some nodes have considerably larger degrees (max 7,000)



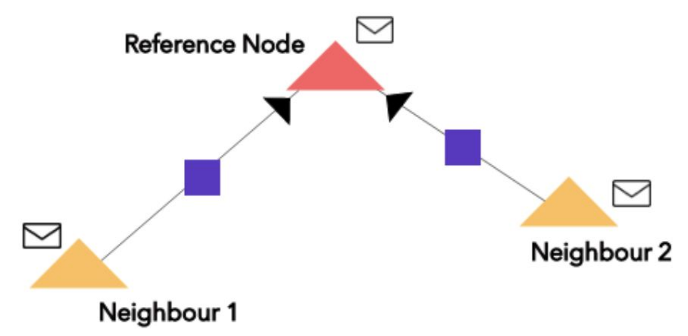


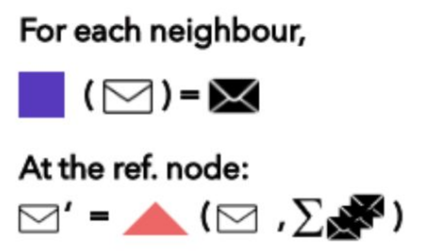
General Approaches to KG Alignment

- Architecture
  a. GCN
  b. GAT (MRAEA)
  c. TransE/MTransE (OTEA & KAGAN)
- Embeddings
  a. Node
  b. Edges
  c. TransE
  d. Attributes

Approach #1: Graph Convolutional Network

- Message Passing:
  - Messages are the node embeddings.
  - For each time step, at each reference node, messages from its neighbors are aggregated (Σ).
  - The aggregated messages becomes the updated embedding of the reference node.





Source: https://medium.com/dair-ai/an-illustrated-guide-to-graph-neural-networks-d5564a551783

Approach #2: Graph Attention Network (GAT)

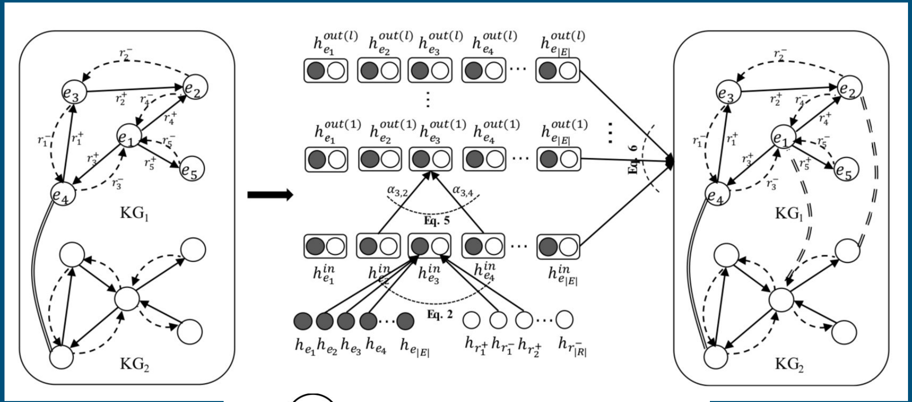
- Assign varying levels of importance to the node's neighborhood.
- A single GAT layer can be described as $e_{ij} = a(\mathbf{W}\vec{x_i}, \mathbf{W}\vec{x_j})$, where
    - e_{ij} - attention coefficient or importance of edge (e_i, e_j)'s features for a source node e_i
    - - graph structure is retained by allowing node *i* to "attend" only to its neighborhood
    - **W** - embedding weights
    - a - any attention function
- The relative attention score is computed using softmax over all the values in the neighborhood.
    - $\vec{x}'_i$ is the transformed node feature of e_i
- GAT employs multi-head attention to stabilize the learning process.

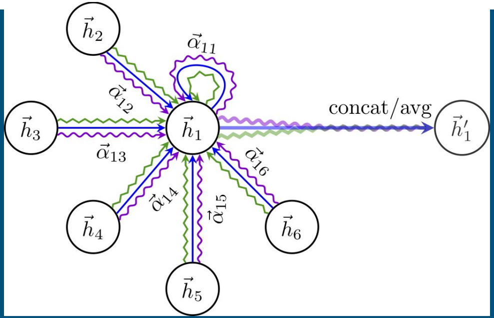
$$\vec{x}'_i = \boxed{\big\Vert_{k=1}^{K}} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{x_j}\right)$$

head softmax normalized attention coefficients of edge(e_i, e_j)

GAT: MRAEA (Mao et al., 2020)

- GAT Model
- Attention Score: node embedding + embeddings of neighbors + relation embeddings (type, direction, inverse - dashed lines)





GAT layer with multi-head attention

Source: https://petar-v.com/GAT/

Approach #3: TransE [Bordes *et al.*, 2013]

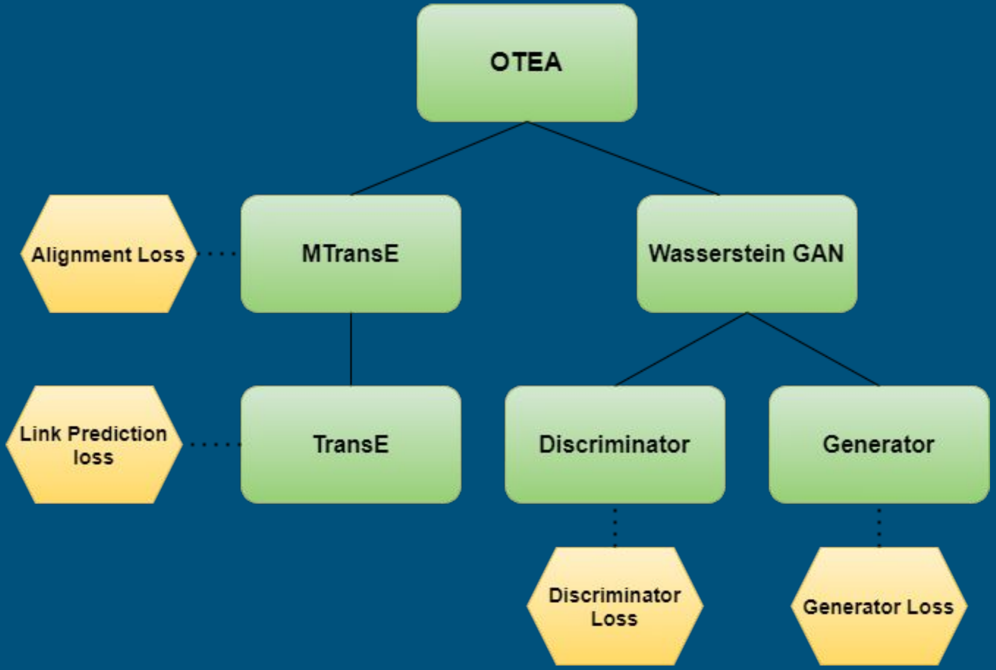
- Embeds (h, r, t) of a KG into a different space
- Goal is that **h** + **r** ≈ **t** holds (boldfaced **h**, **r**, **t** are the embedded h, r, t respectively)

Approach #3: MTransE [Chen *et al.*, 2016]

- 2 KGs are embedded using TransE
- A linear transformation between the spaces is learned via L2-Loss

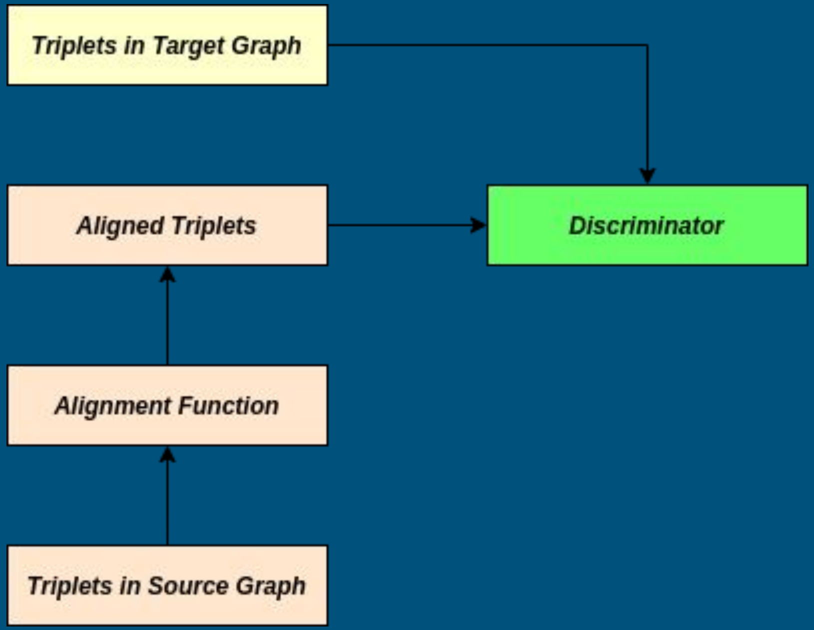
OTEA (Optimal Transport Entity Alignment)

- Extends MTransE [Chen et al., 2016]
- Additionally considers Group-Level Loss (distance between E and M´E´)
- Group-Level Loss computed using Wasserstein GAN [Arjovsky et al., 2017]



KAGAN

- Based on MTransE
- Generator creates fake examples
- Discriminator minimizes difference btw. the *aligned triplets* & *triplets in target* Graph



DBS Framework Adaptation

DBS Modules	OTEA	KAGAN	MRAEA
Dataset Loader	✓	✓	✓
Embeddings	+ (MTransE)	+ (MTransE)	✎ (Relation)
Graph (Message Passing	✓	✓	✓
Layers	+ (GAN)	+ (GAN)	+ (GAT)
Similarity	✓	✓	✓
Trainer	✓	✓	✎ (Bootstrap)
Evaluator	✓	✓	✓

✓: Adapted as is ✎: Revised +: New N/A: Not Applicable

Results - OTEA

	Wk-31-15k Dataset	en-fr			en-de		
		Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR
1.	Results in Paper	0.375	0.574	0.472	0.374	0.572	0.470
2.	Published Code Results	0.371	0.465	0.420	0.278	0.352	0.326
3.	DBS Framework	0.080	0.156	0.149	0.113	0.245	0.184

Results - MRAEA

DBP15k_JAPE (zh_en)		Hits @ 1 (%)	Hits @ 5 (%)	MRR
1.	Results in Paper	75.28	92.31	0.824
2.	Published code Results	62.56	82.34	0.715
3.	DBS Framework	11.18	40.27	0.206
4.	- 49k Epochs	19.55	52.67	0.303
	Difference (#2 - #4)	43.01	29.77	0.412

"All parameters being equal…"

emb_dim	: 100
margin loss	: 3
layers	: 2
heads	: 2
emb_dropout	: 0.3
Adam lr	: 0.005
bias	: yes
train-test split	: 0.30
eval split	: None
batch_size	: num_entities
epochs	: 5000

Insights & Challenges

- Steep learning curve when using the DBS Framework
- Difficult to implement approach just by looking at the paper description (formula, etc.)
    - Some had no published code.
    - Some published code required debugging.
    - Published code included details that were not described in the paper.
- Difficult to work in the group because we were assigned different papers

Future Work

- *Find out why results differ from the published results*
- *DBS Framework documentation*
    - class organization, inventory of methods
- *Continue on different approaches*

Thank You!

Questions?