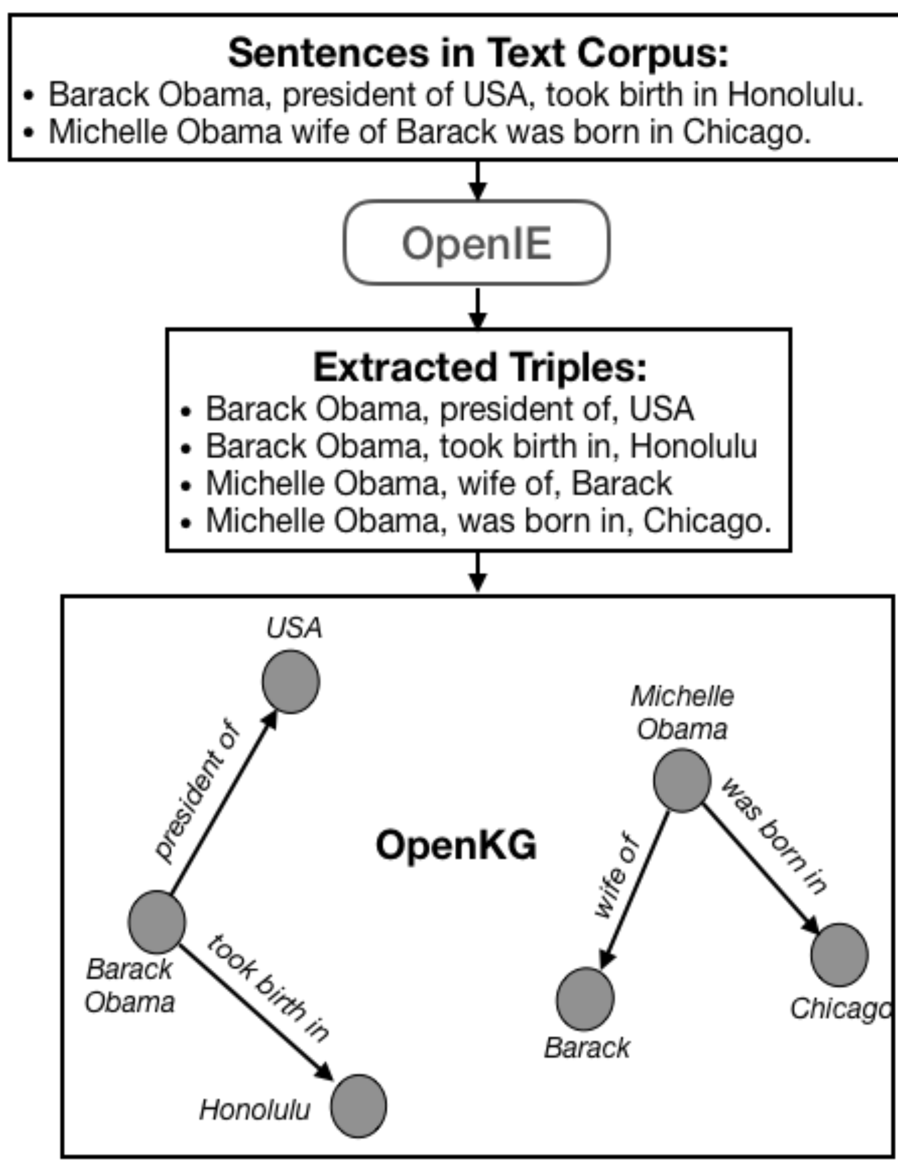
CaRe: Open Knowledge Graph Embeddings

Swapnil Gupta[1], Sreyash Kenkre[2], Partha Talukdar[1]

[1]Indian Institute of Science, Bangalore [2]Walmart Labs, Bangalore

Open Knowledge Graph (OpenKG)



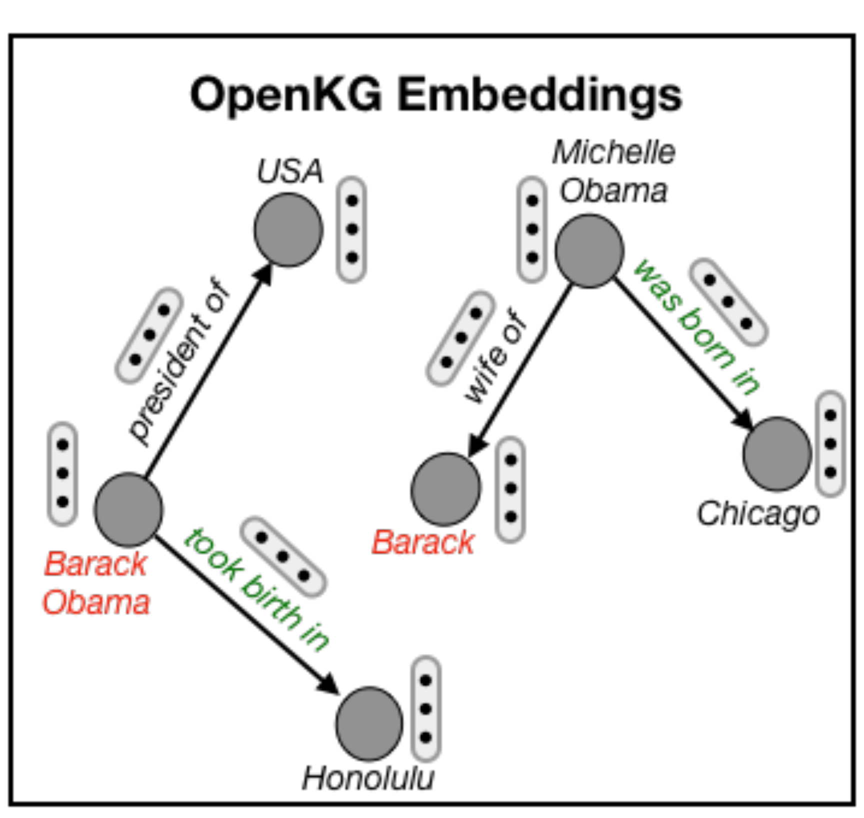
Open Information Extraction (OpenIE): Automatically extracts *(subject phrase, relation phrase, object phrase)* triples from a text, e.g., *(Barack Obama, took birth in, Honolulu).*

OpenKG: Noun Phrases (NPs) are nodes and relation phrases (RPs) are edge labels.

Advantages: Don't require human supervision and pre-specified ontology making them highly adaptable.

OpenKG Embeddings and Challenges

OpenKG Embeddings: Learning vector representations of the NPs and RPs in the graph, numerically encoding the facts in the graph.

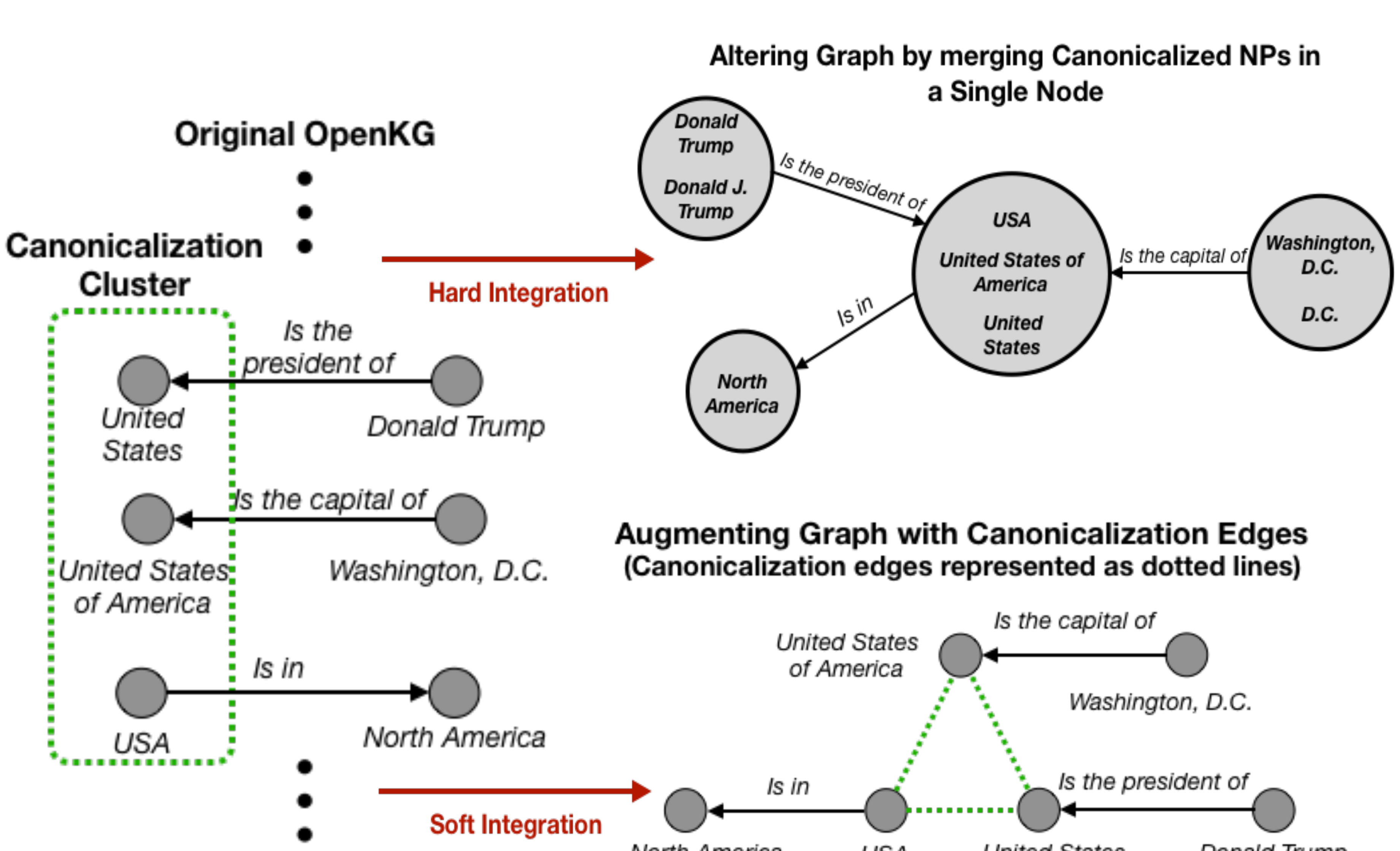


Challenges

- Same latent entity and semantic relation represented with different NPs and RPs respectively(e.g., *Barack Obama* and *Barack*, *was born in* and *took birth in*).
- Fragmented and Sparse graph.
- Existing *ontological* KG embedding methods assume each node a unique entity and each edge label a unique relation, hence not suitable for OpenKG embeddings.

Densifying OpenKG with Canonicalization

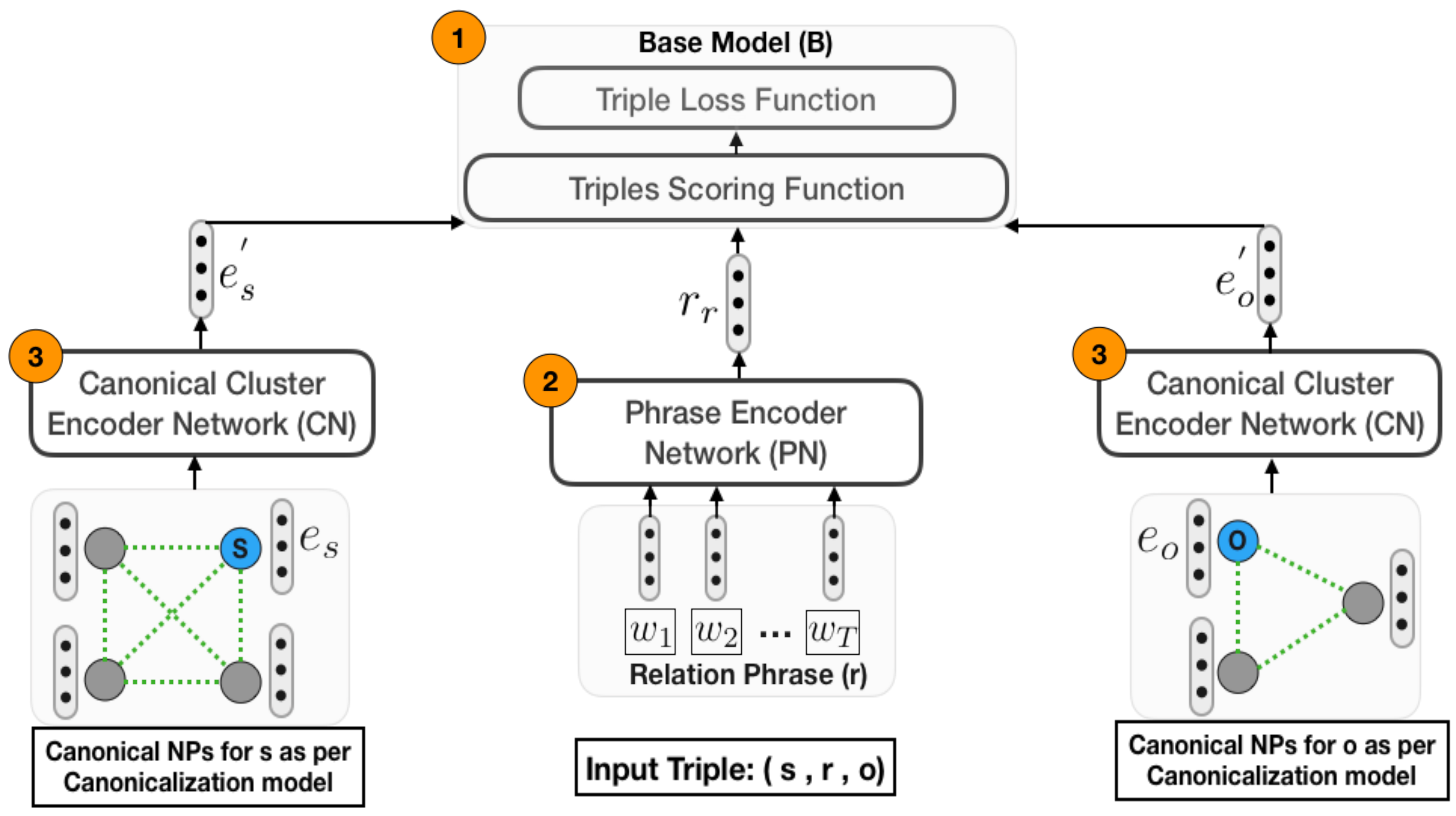
Canonicalization: Clustering NPs referring to the same entity and RPs having the same semantics (CESI (Vashishth et al., 2018)).



Shortcomings of Hard Integration

Automatic canonicalization models often output some incorrectly canonicalized elements. Thus, directly merging nodes would result in propagation of errors.

CaRe Architecture



1. **Base Model (B):** A KG embedding model (e.g., TransE, ConvE)
2. **Phrase Encoder Network (PN):** A sequence encoder network. (Default: PN = *Bi-GRU with last pooling*)
3. **Canonical Cluster Encoder Network (CN):** Differentiable message passing network. (Default: CN = *Local Averaging Network (LAN)*).

CaRe infuses canonicalization information by operating on the densified OpenKG. Parameterizing RP embeddings allows parameter sharing through word overlap.

Results

Dataset Statistics

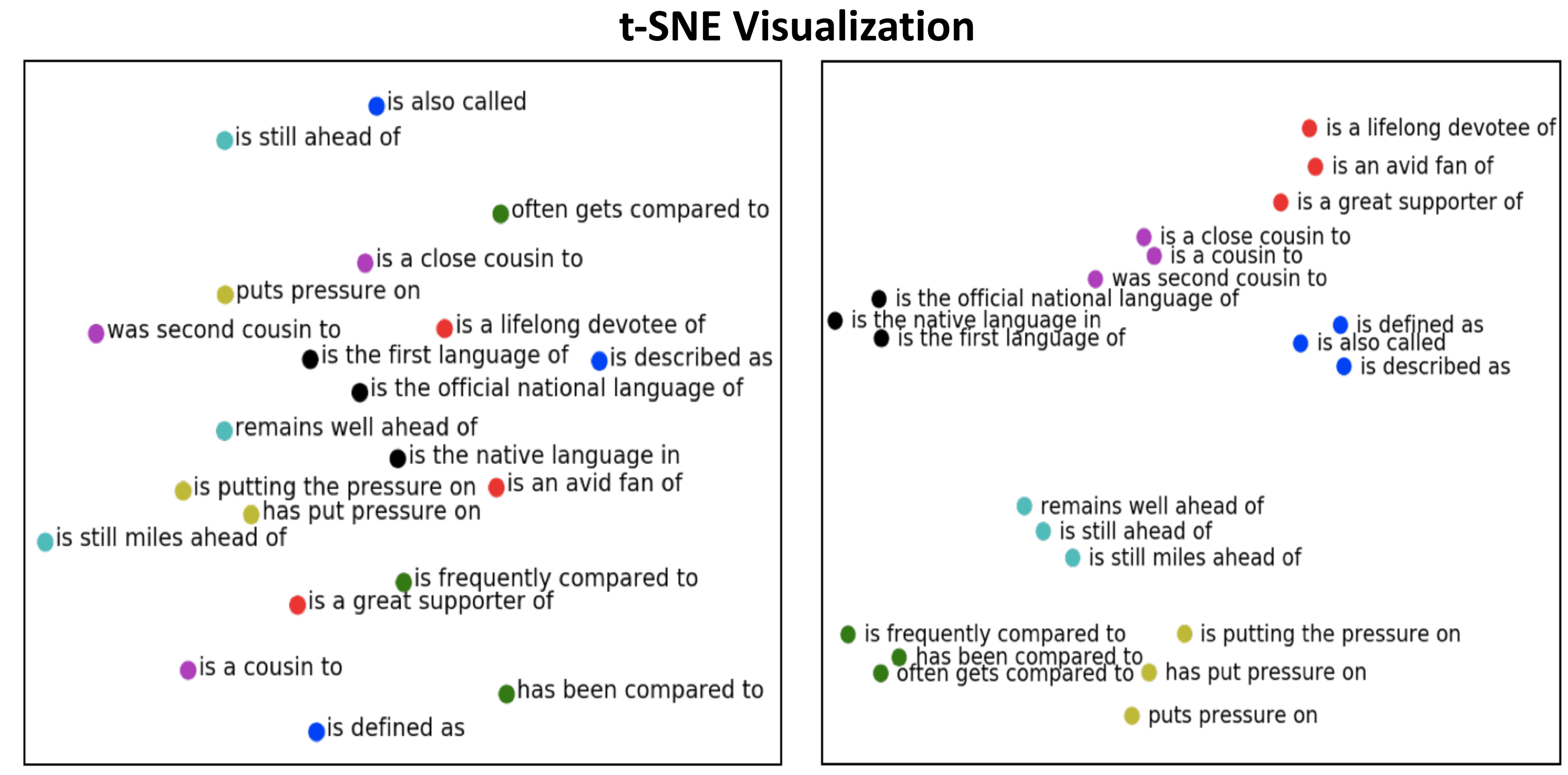
ReVerb45K – #NPs: 27K, #RPs: 21.6K, #Triples: 45K
ReVerb20K – #NPs: 11.1K, #RPs: 11.1K, #Triples: 19.5K

Link Prediction results: Predicting missing links in the knowledge graph.

Method	ReVerb45K					ReVerb20K				
	MR	MRR	Hits@10	Hits@30	Hits@50	MR	MRR	Hits@10	Hits@30	Hits@50
TransE	2955.8	.193	.361	.446	.478	1425.8	.126	.299	.411	.468
TransH	2998.2	.194	.362	.442	.478	1464.4	.129	.303	.409	.467
DistMult	8988.8	.051	.051	.052	.065	6260.0	.033	.044	.055	.060
ComlEx	7786.5	.047	.047	.048	.073	5502.2	.037	.058	.075	.085
R-GCN	2866.8	.042	.046	.091	.113	1204.3	.122	.187	.263	.305
ConvE	2650.8	.233	.338	.401	.429	1014.5	.294	.402	.491	.541
CaRe(B=ConvE)	**1308.0**	**.324**	**.456**	**.543**	**.579**	**973.2**	**.318**	**.439**	**.525**	**.566**

Superior performance of CaRe supports the hypothesis that, infusion of canonicalization information while learning OpenKG embeddings is beneficial.

t-SNE Visualization



ConvE CaRe(B=ConvE)

Parameterizing RP embeddings helps in capturing the semantic similarity in CaRe.

Acknowledgement

MHRD
Government of India
Ministry of Human Resource Development

Source Code

Source: *github.com/malllabiisc/CaRE*

Contact: *swapnilgupta.229@gmai.com*