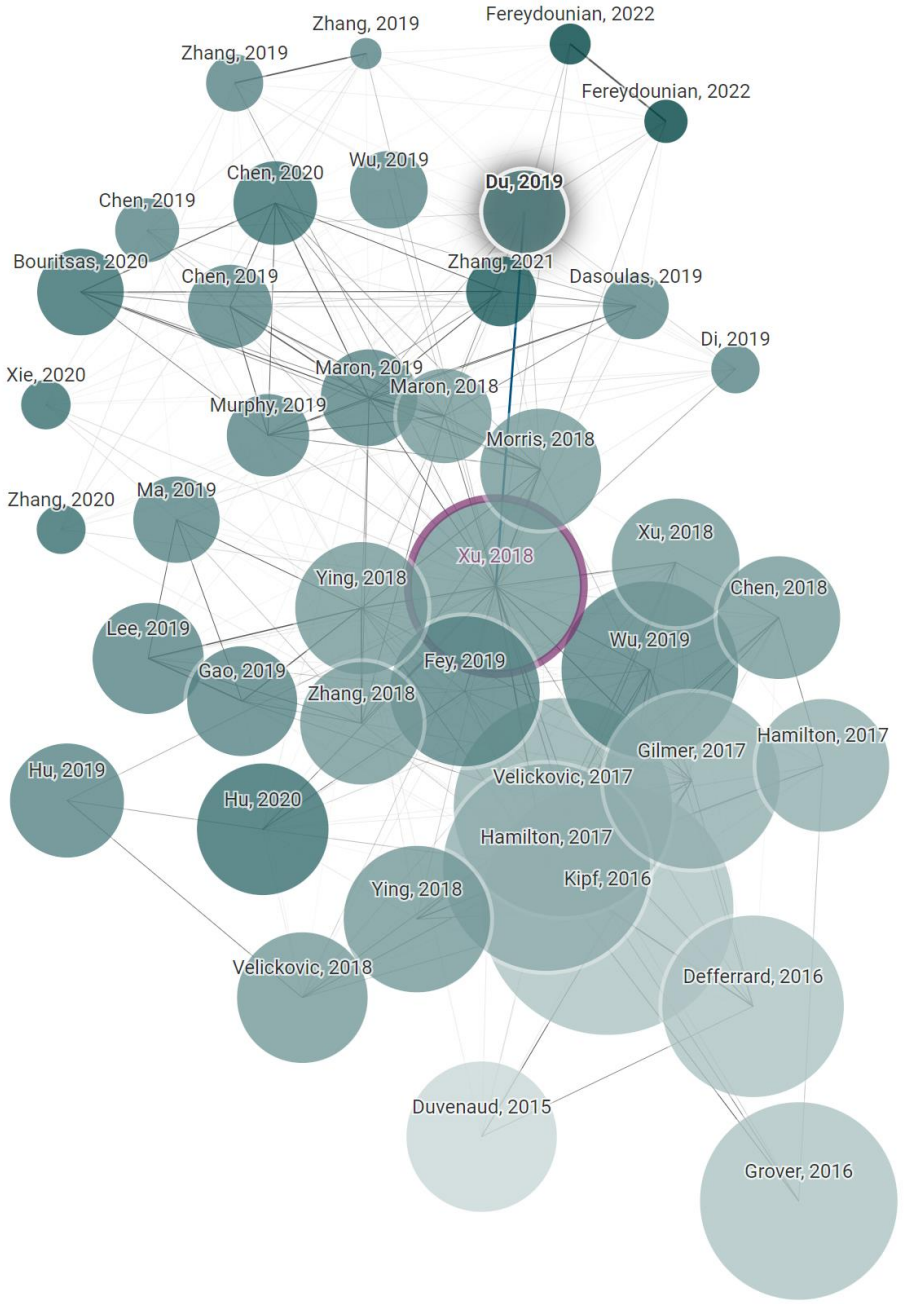
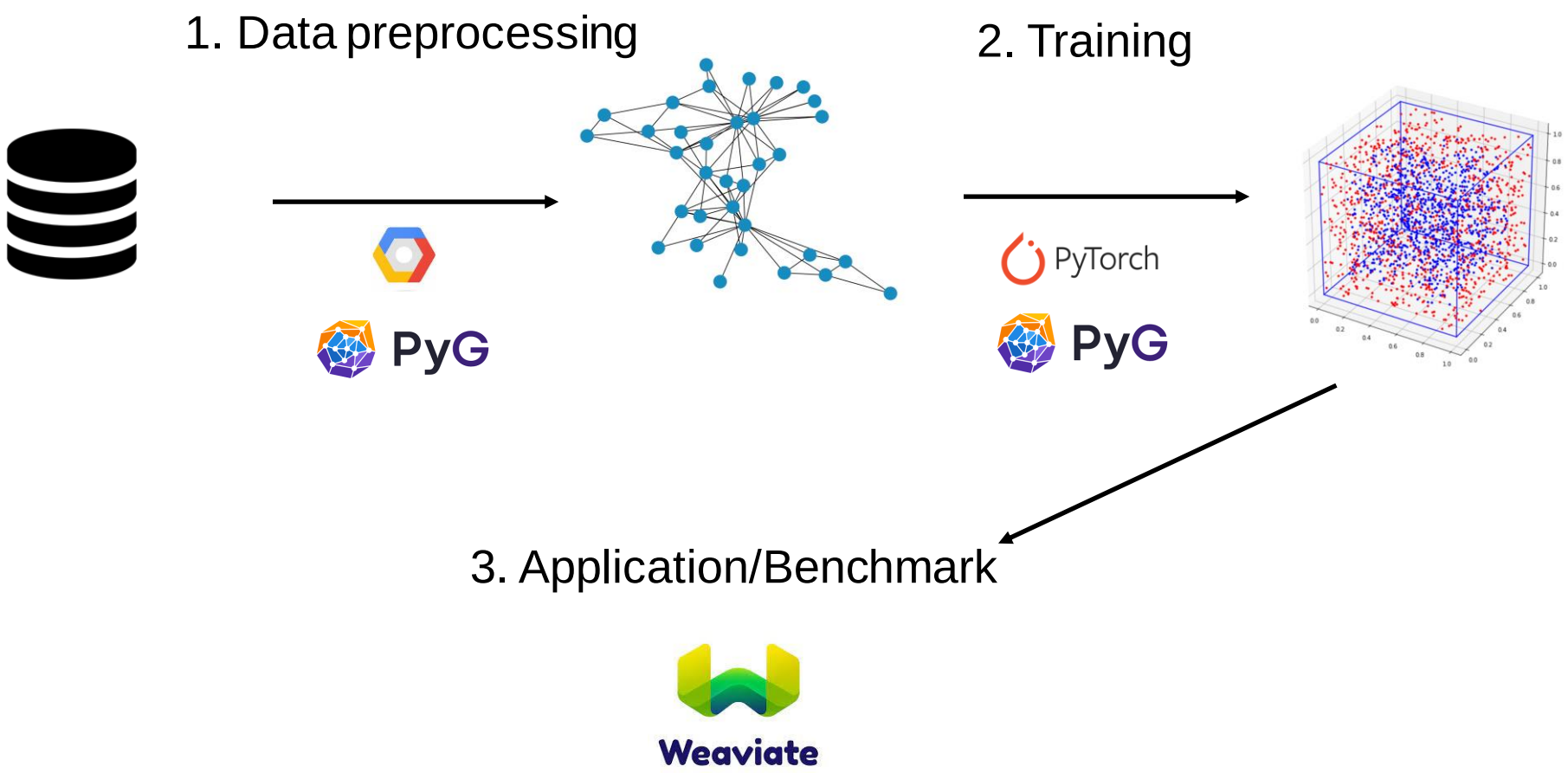
Semantic Document Search with Graph Neural Networks

Modul Aktuelle Data Science-Entwicklungen

Amos Dinh, Ahmet Korkmaz,
Henrik Rathai, Matthias Fast

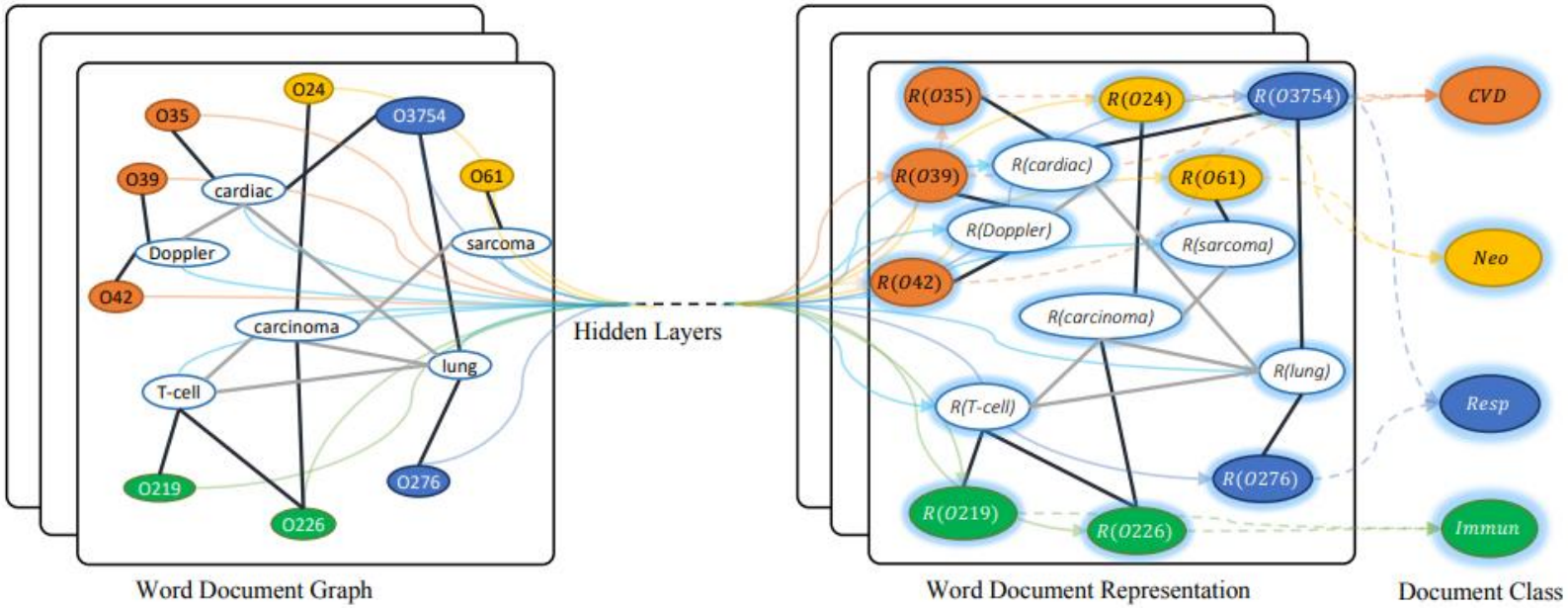


Overview


1. Data preprocessing
2. Training
PyG
PyTorch
PyG
3. Application/Benchmark
Weaviate

Reference Approach

"Graph convolutional networks for text classification"

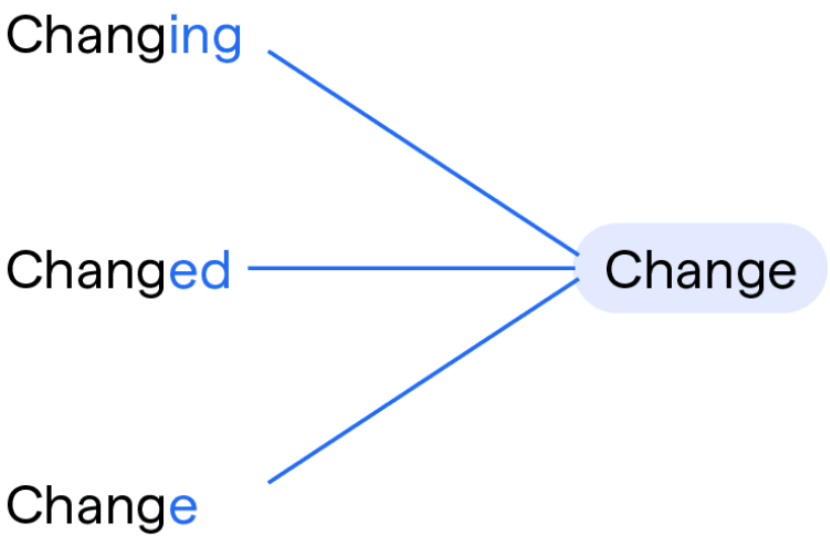


L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in Proceedings of the AAAI conference on artificial intelligence (2019), Vol. 33, pp. 7370–7377

Dataset

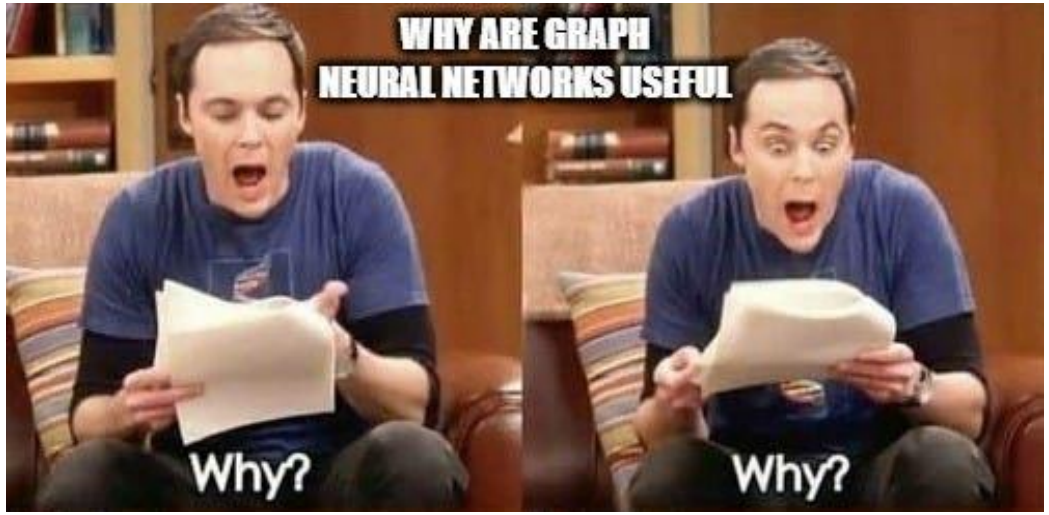
e.g. id, title, comments (pages), journal-ref, doi, categories, license, versions (created), authors, abstract

Lemmatization and removing stopwords



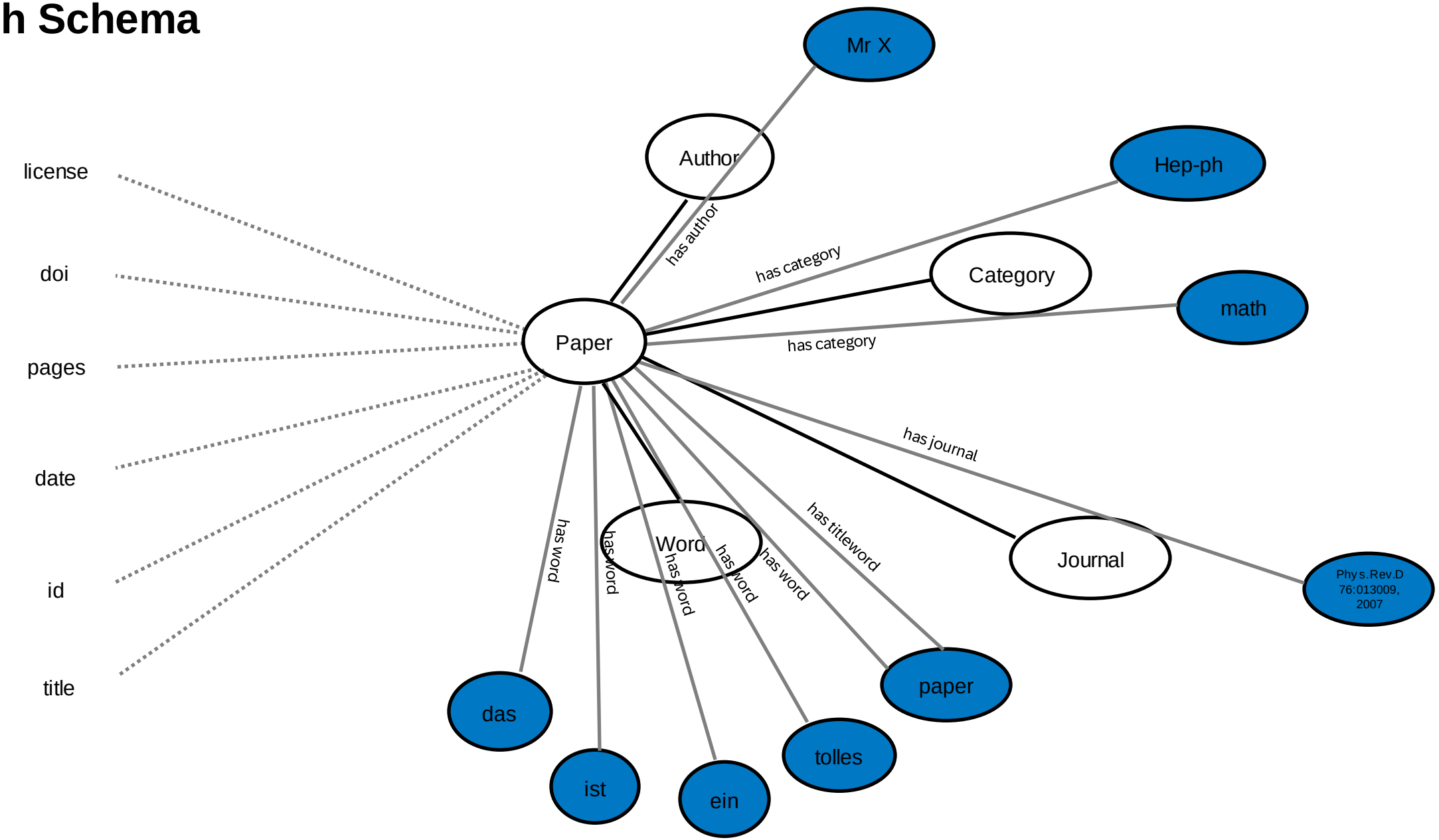
When was the first computer invented?
How do I install a hard disk drive?
How do I use Adobe Photoshop?
Where can I learn more about computers?
How to download a video from YouTube
What is a special character?
How do I clear my Internet browser history?
How do you split the screen in Windows?
How do I remove the keys on a keyboard?
How do I install a hard disk drive?

Meme of the day part I

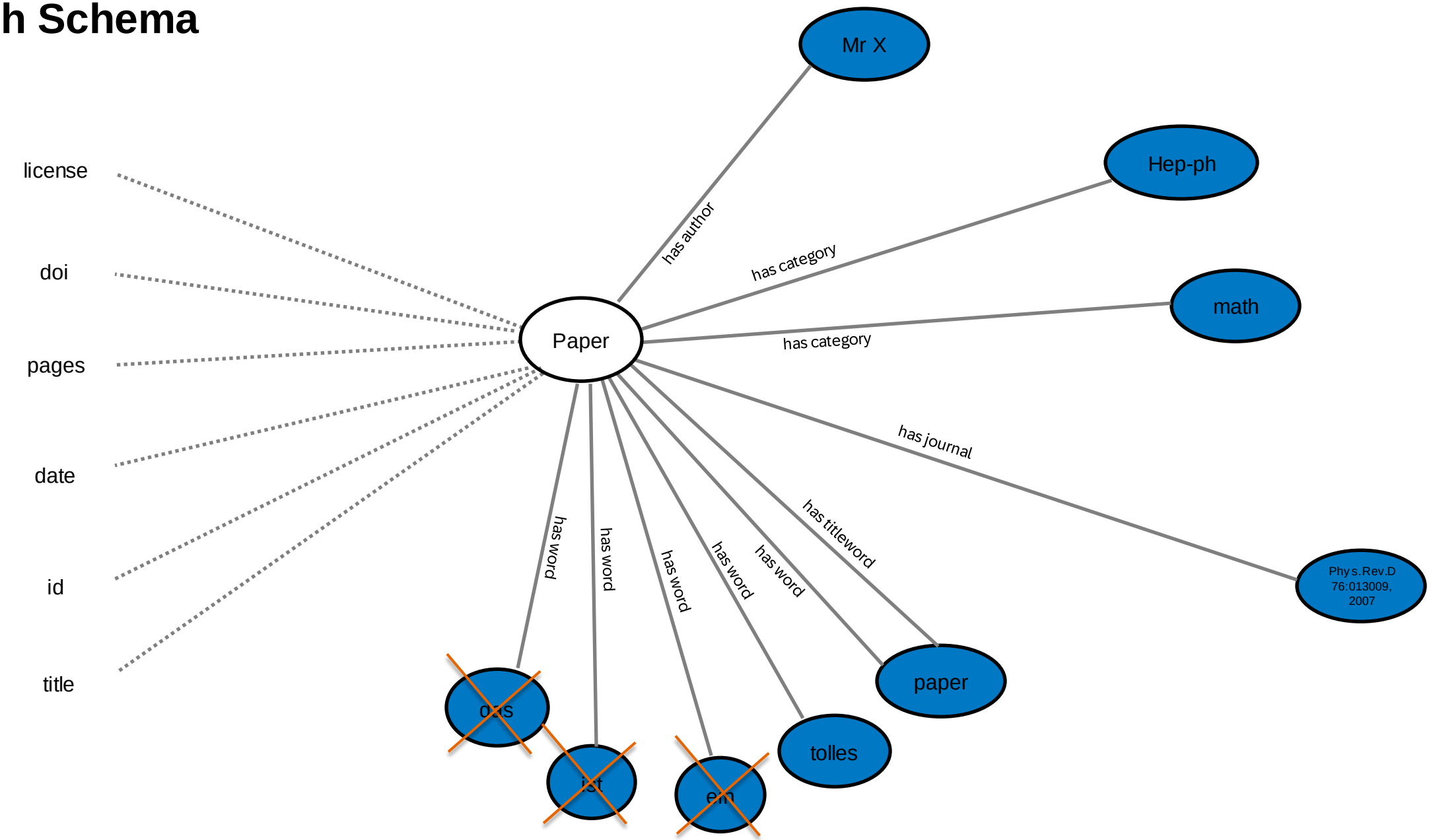


https://medium.com/sfu-cspmp/an-overview-graph-neural-networks-b071ce1739fd

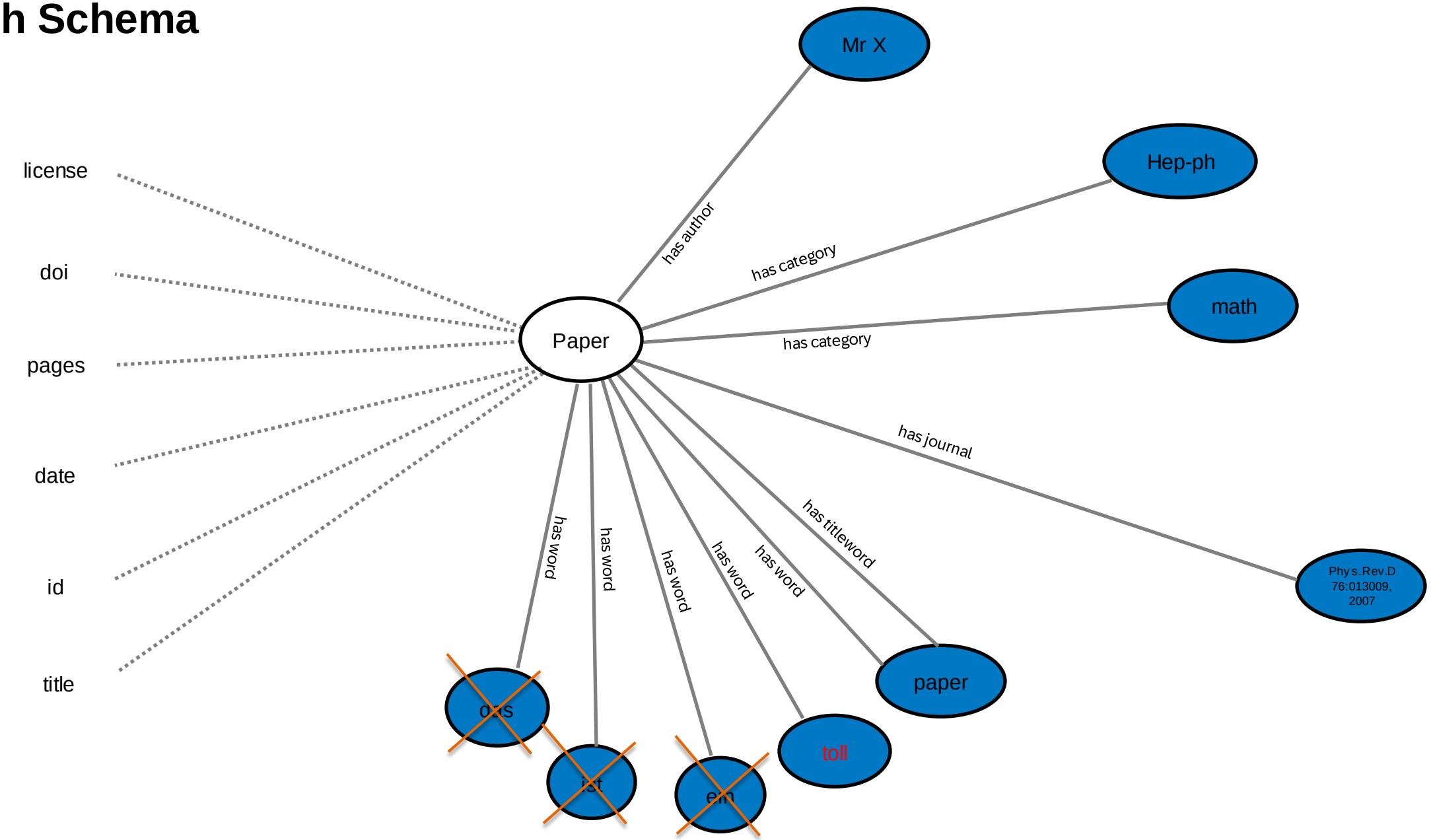
Graph Schema



Graph Schema



Graph Schema



Graph Modeling Edges – TF-IDF

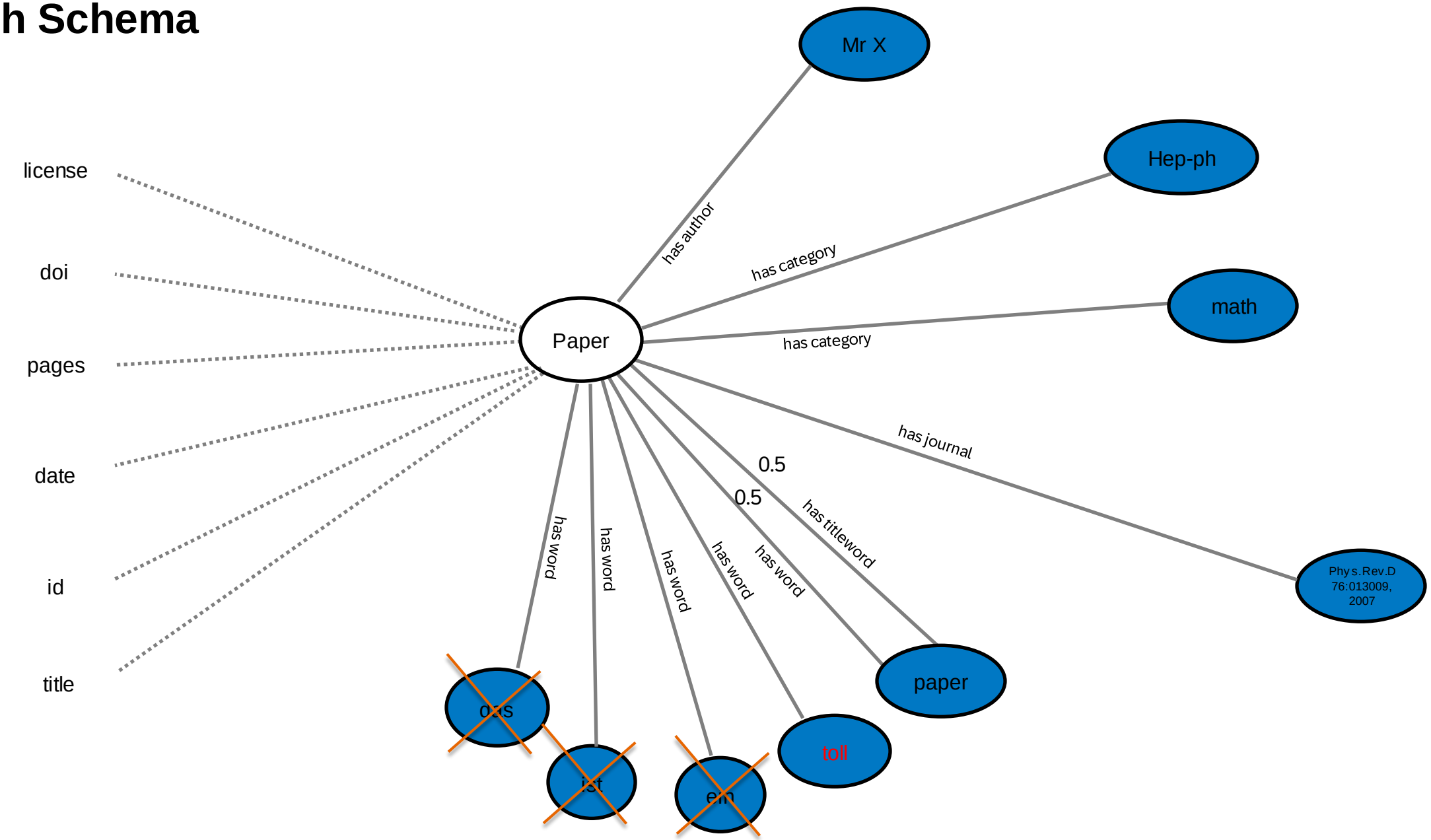
Term Frequency – Inverse Document Frequency

TF (term frequency) = count of a word in a document

$$IDF(t) = \ln(\frac{1+n}{1+df(d,t)})+1$$

sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.4.0 documentation

Graph Schema



Graph Modeling Edges - PMI

Pointwise Mutual Information

$$PMI(x,y) = \ln\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Bouma, Gerlof J.. “Normalized (pointwise) mutual information in collocation extraction.” (2009).

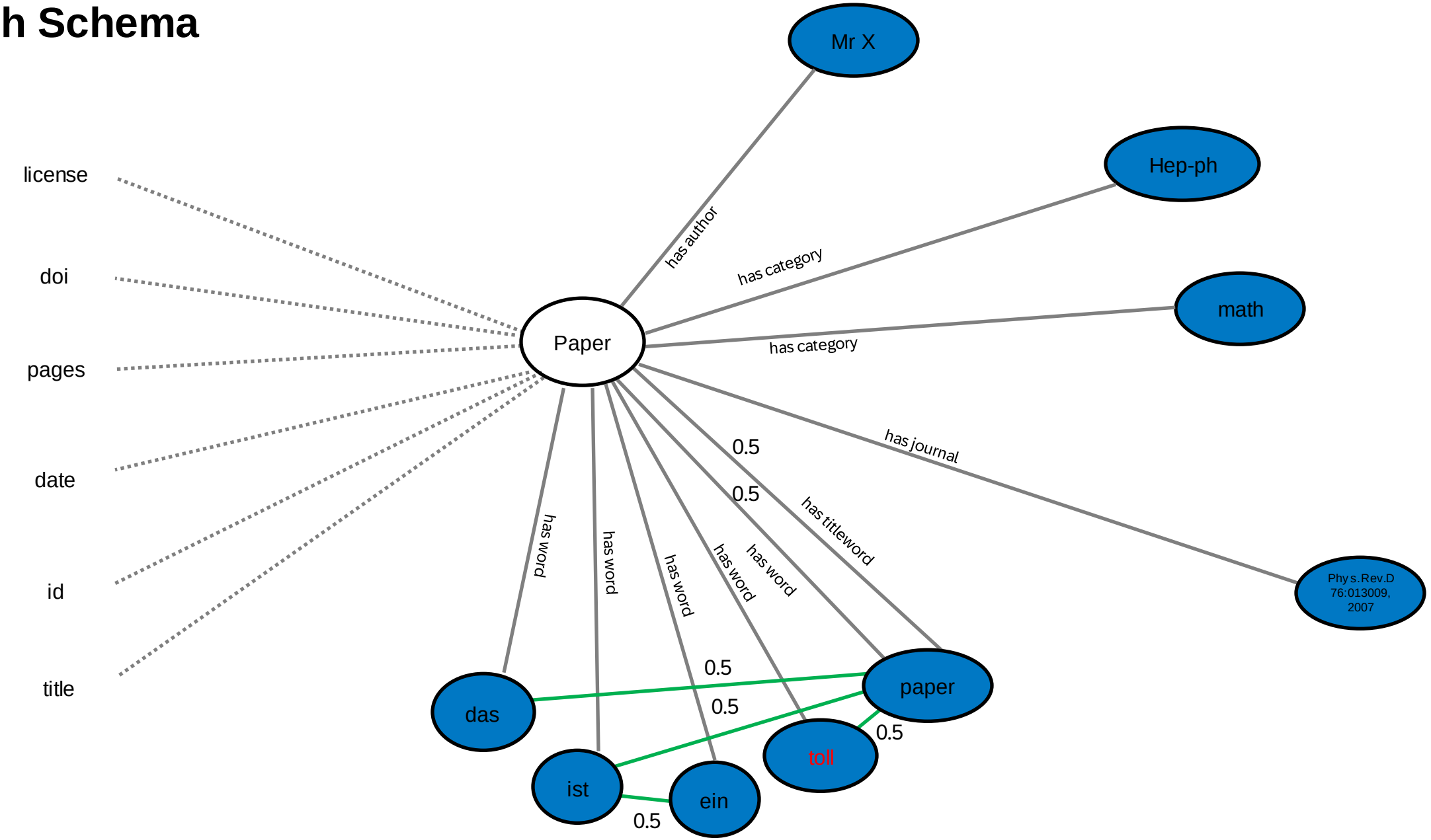
Graph Modeling Edges - NPMI

Normalized Pointwise Mutual Information

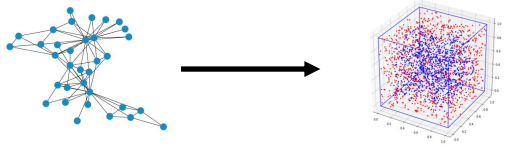
$$NPMI(x,y) = \frac{\ln(\frac{p(x,y)}{p(x)p(y)})}{-\ln p(x,y)}$$

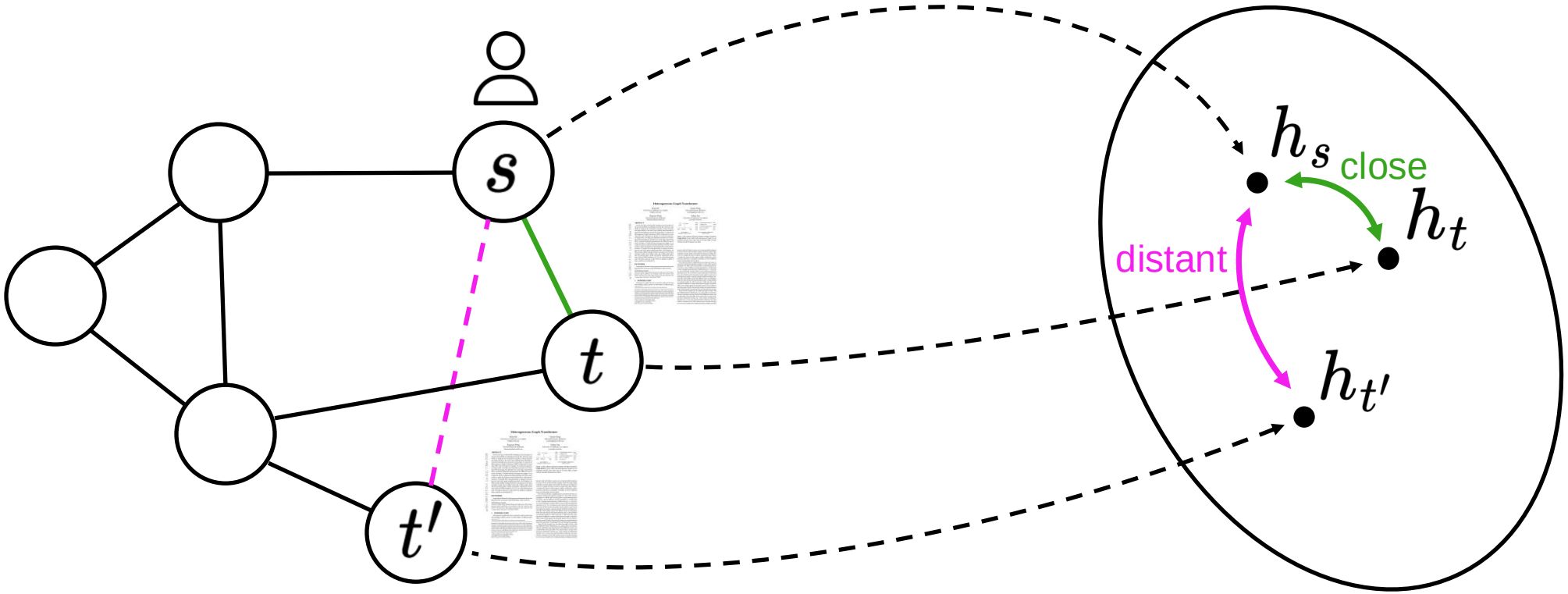
Bouma, Gerlof J.. “Normalized (pointwise) mutual information in collocation extraction.” (2009).

Graph Schema



Training



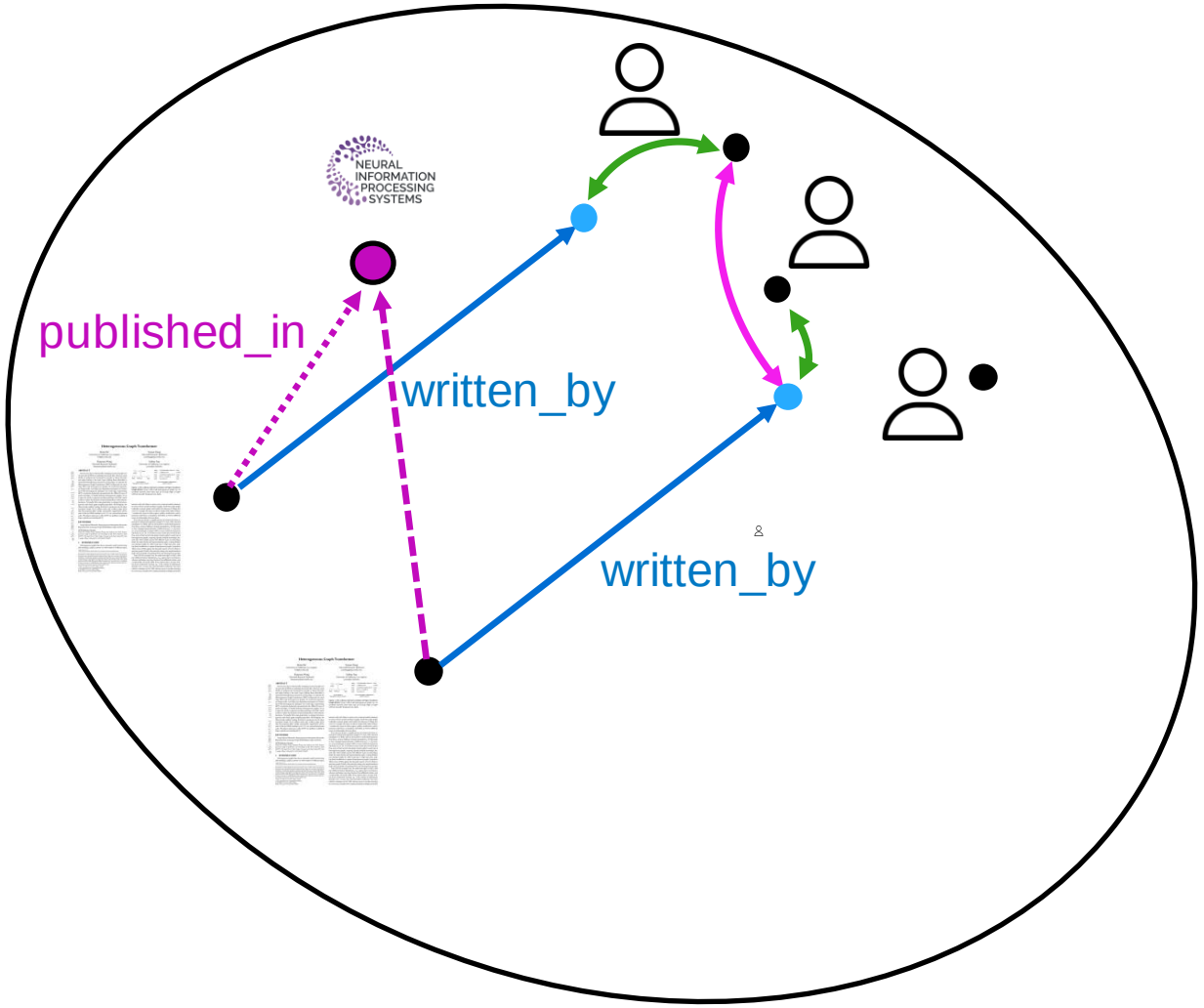


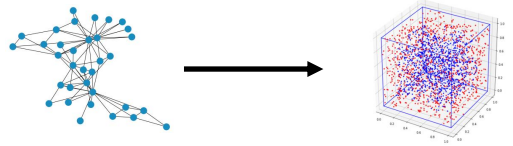
training triplet

$$\begin{bmatrix} h_s \\ h_t \\ h_{t'} \end{bmatrix}$$

arXiv-paper graph → System Graph Neural Network → **embedding space**

Training: TransE



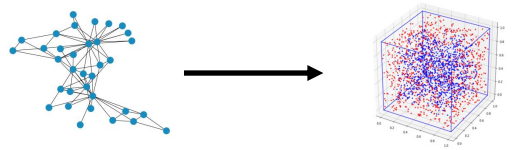


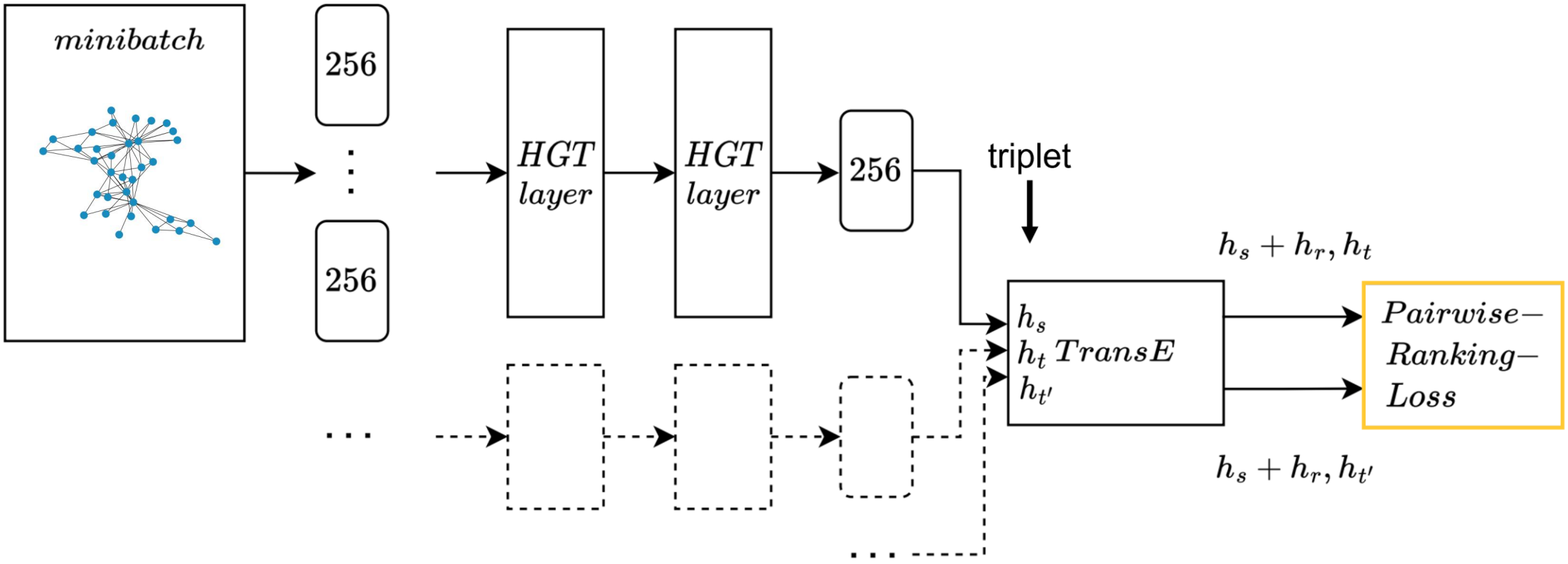
$$d(h_s + h_r, h_t)$$

+ written_by =

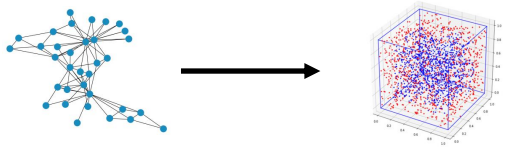
Bordes, Antoine, et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems* 26 (2013).

Training




minibatch
256
256
HGT layer
HGT layer
256
triplet
h_s
h_t TransE
$h_{t'}$
$h_s + h_r, h_t$
$h_s + h_r, h_{t'}$
Pairwise–Ranking–Loss

Training

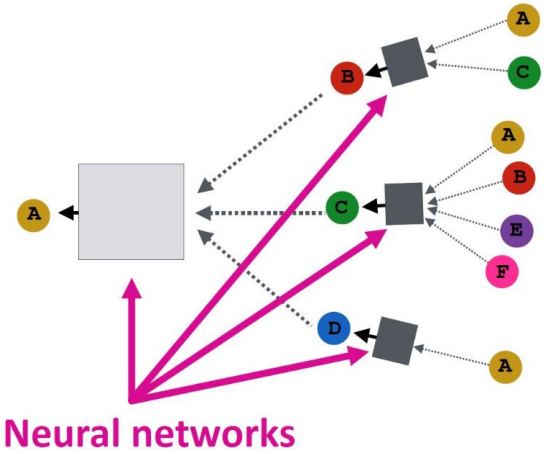


- Training with approximately 1mio edges (+ negatives)
  - Seen nodes (~ seen edges): 1mio * (2 + 256 + 2024) = 2,2mrd
  - 36 hours on P100 GPU
- Heterogenous Graph Transformer and heterogenous graph sampling

$$ATT\text{-}head^i(s,e,t) = \left(K^i(s)\, W^{ATT}_{\phi(e)}\, Q^i(t)^T\right) \cdot \frac{\mu_{\langle \tau(s),\phi(e),\tau(t)\rangle}}{\sqrt{d}}$$

$$K^i(s) = \text{K-Linear}^i_{\tau(s)}\left(H^{(l-1)}[s]\right)$$

$$Q^i(t) = \text{Q-Linear}^i_{\tau(t)}\left(H^{(l-1)}[t]\right)$$



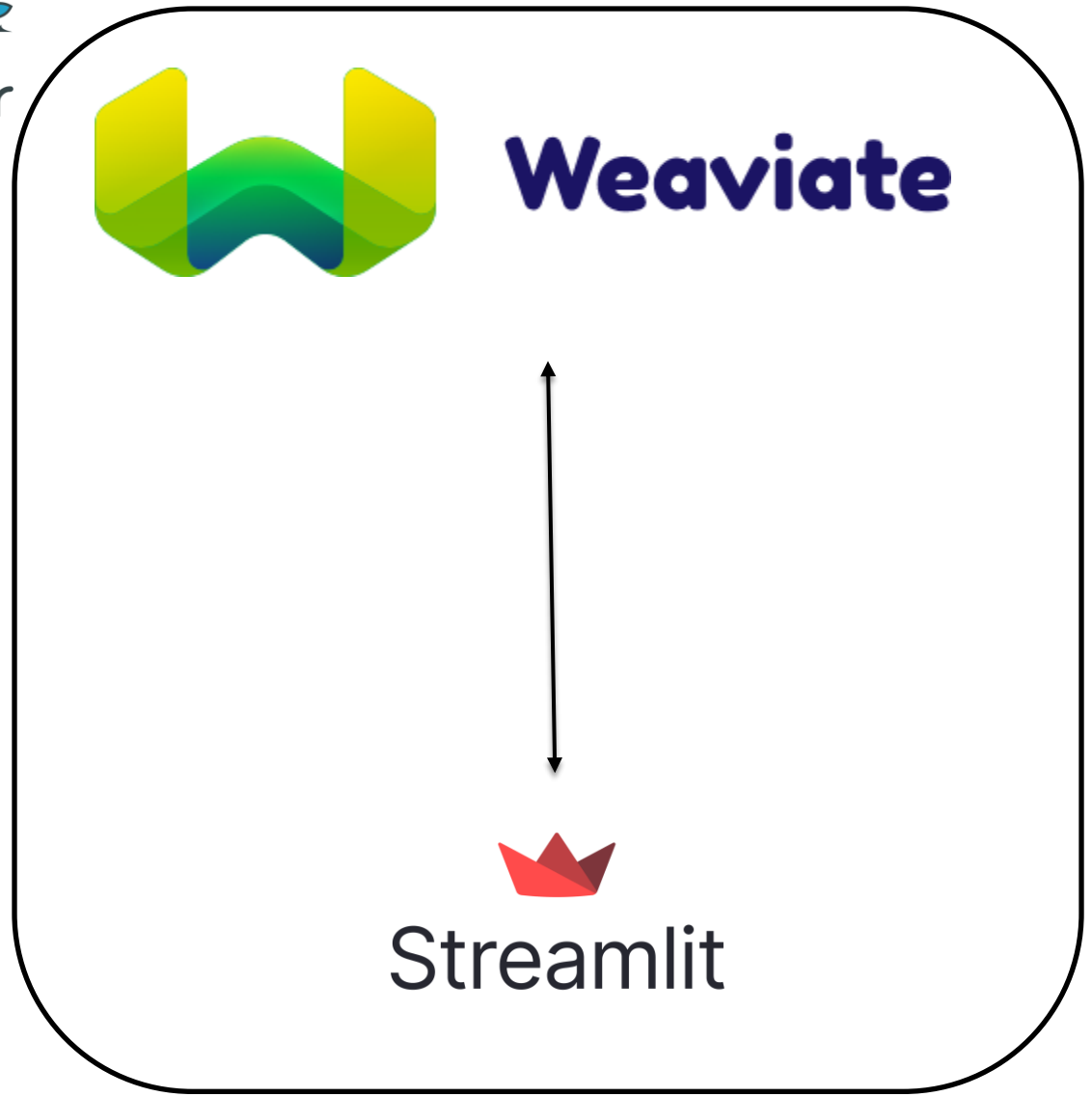
Hu, Ziniu, et al. "Heterogeneous graph transformer." *Proceedings of the web conference 2020.*

Challenges

- Size of the dataset
  - Preprocessing
  - Training
  - Implementation

Architecture

- Build as a Docker compose file
- Frontend server
- Weaviate vector database



Benchmarking results: Comparing TF-IDF and proposed Method

- Based on limited benchmarking data (our own, independent of any training)
- Average Rank (per query title) (lower is better)

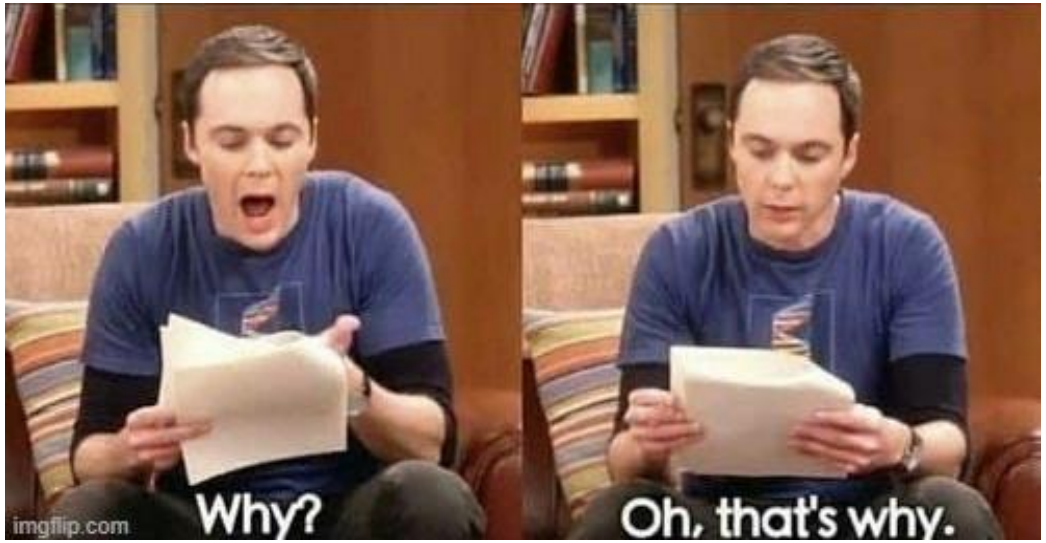
TF-IDF: **189037**/2381173

Proposed Method: **81752**/2381173

```
arXiv:1801.07606,Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning,arXiv:2008.09864,Tackling Over-Smoothing for General Graph Convolutional Networks
arXiv:1801.07606,Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning,arXiv:2002.05287,Geom-GCN: Geometric Graph Convolutional Networks
arXiv:1810.00826,How Powerful are Graph Neural Networks?,arXiv:1810.02244,Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks
arXiv:1709.05254,Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks,arXiv:1908.00734,Detection of Accounting Anomalies in the Latent Space using Adversarial Autoencoder Neural Networks
arXiv:1709.05254,Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks,arXiv:2210.14056,Unsupervised Anomaly Detection for Auditing Data and Impact of Categorical Encodings
arXiv:1803.01092,Analyzing Business Process Anomalies Using Autoencoders,arXiv:1908.00734,Detection of Accounting Anomalies in the Latent Space using Adversarial Autoencoder Neural Networks
arXiv:2203.16060,Understanding Graph Convolutional Networks for Text Classification,arXiv:1809.05679,Graph Convolutional Networks for Text Classification
```

LIVE DEMO

Meme of the day part II



https://medium.com/sfu-cspmp/an-overview-graph-neural-networks-b071ce1739fd