Graph Embedding

NLP-BA-Seminar

Noah Hurmer ● noah.hurmer@campus.lmu.de ● 08.07.2021

Outline

- Data structure Graph
- Motivation
- Embedding Types & Applications
- Techniques
- node2vec
- Graph embedding in NLP
- Outlook use cases

What is a Graph?

- Broadly defined and versatile Data structure
- Network of Entities and pairwise Relations

$$G = (V, E)$$

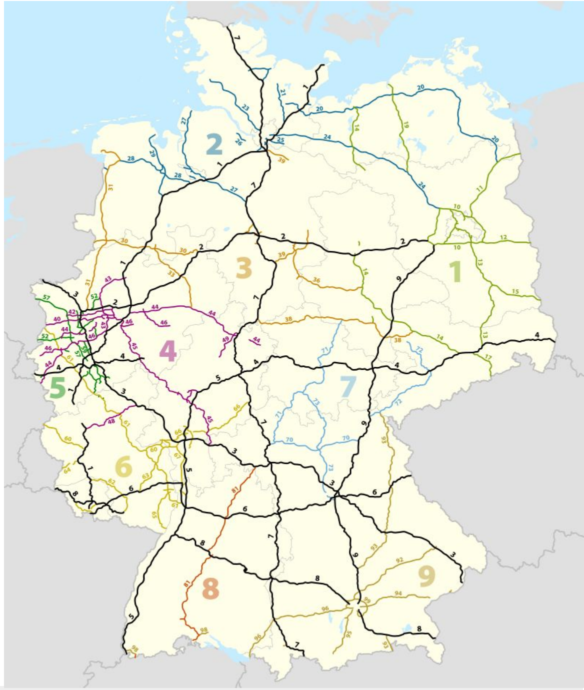
Vertices (Nodes) Edges

What is a Graph? II

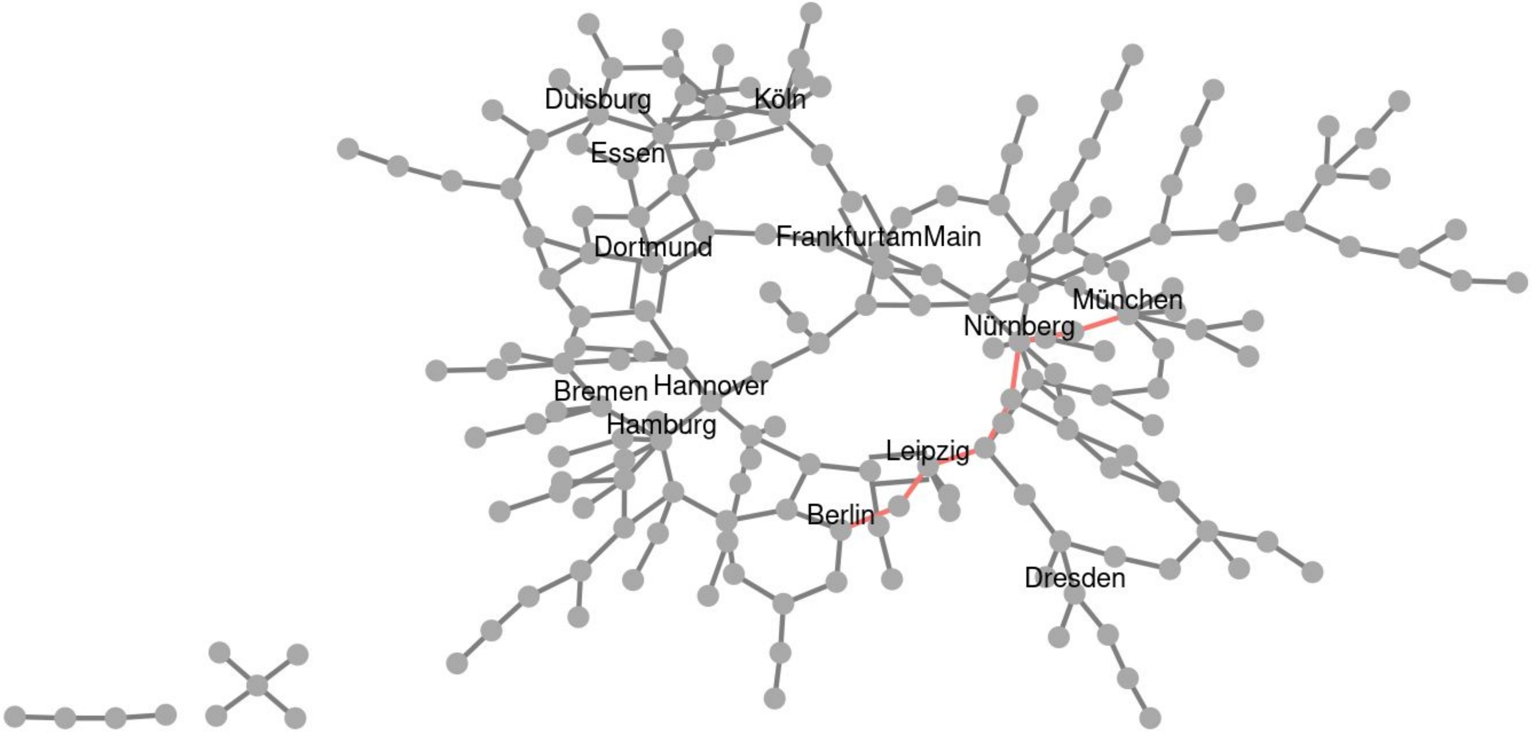
Example

- Network of German Autobahnen
  - A1-99
  - > 65km
- 188 nodes
- 225 edges

figure 1: NordNordWest, 2017



What is a Graph? II



What is a Graph? II

Nodes

- Entity
- Any type of data (Profiles, Images, text, etc)

Edges

- Any type of relationship (Connection, Similarity, Is A, interaction, etc)
- Directional
- Weighted

What is a Graph? II

Additional Graph modes

- Additional Attributes
- Heterogeneous
- Semantic (Knowledge Graph)

Why embed?

Problems

- Storage and computational cost
  - edgelist $|E|$
  - adjacency Matrix $|V \times V|$
- Limited Analysis tools
  - non euclidean space
  - few applicable algorithms: A*, Dijkstra

Why embed? II

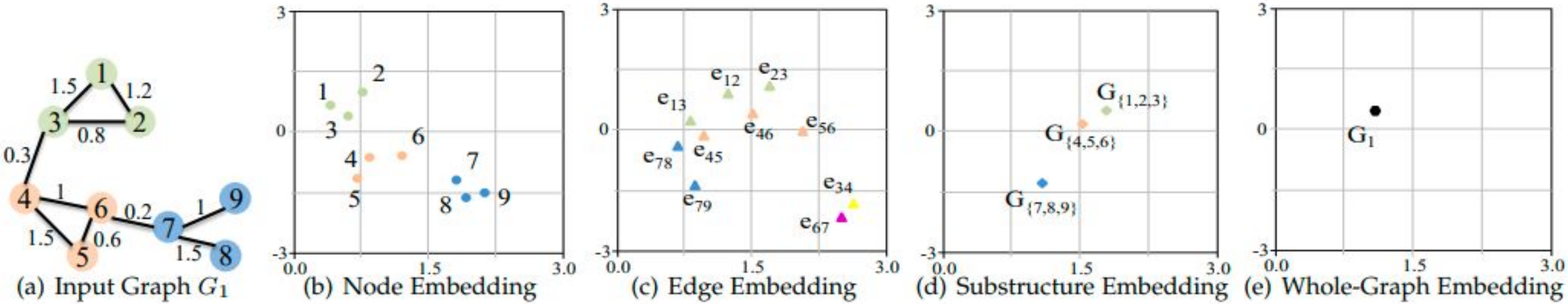
Embedding into low dimensional feature space

- compress information of Graph
- widen spectrum of tools
- increase performance
- remove / reduce manual feature engineering

Embedding Types

- Node
- Edge
- Whole graph
- Subgraph / hybrid

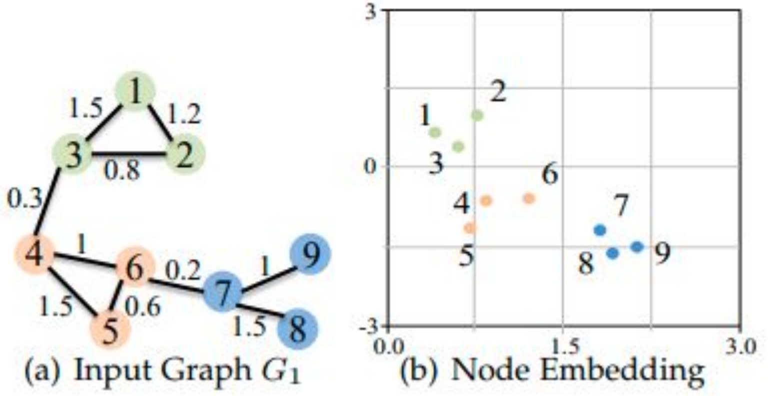
figure 2: Cai et al. 2018



Node Embedding

- Every node to a vector of $\mathbb{R}^d$
- Preserve (structural) “proximity” of nodes
  - define proximity
  - similarity between neighbourhoods
- Applications
  - Compression
  - Node classification
  - Node similarity & Clustering

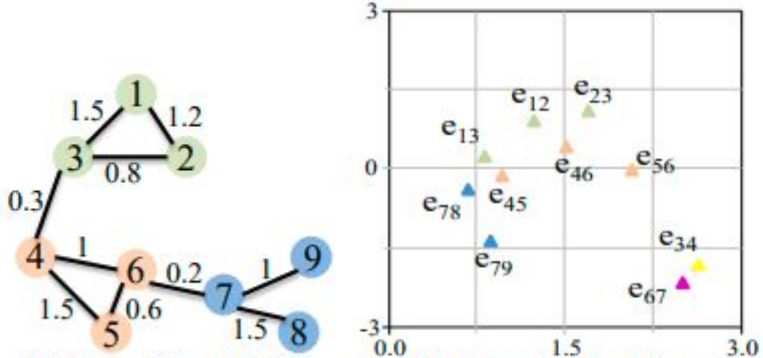
figure 2: Cai et al. 2018



Edge Embedding

- Every edge to a vector of $\mathbb{R}^d$
- Embed node pair or triplets $< a, r, b >$
- Preserve relations between nodes
- Applications
  - Link prediction
    - suggestions
  - Attributed edges
    - social networks

figure 2: Cai et al. 2018



(a) Input Graph G_1 (c) Edge Embedding

Other

- Whole Graph
  - compare graphs
  - Proteins
- Hybrid / Subgraph
  - combines types to embed graphlets, community
  - semantic search, classification

Embedding techniques

- Matrix factorization
- Deep Learning
  - DL with Random Walks
- Edge reconstruction
- Graph kernel
- Generative model

node2vec

- Node embedding model
- Utilizes 2nd order Random Walk
- DL - model SkipGram
- Extension on DeepWalk
    - Adaptation of word2vec

Random Walk

- Efficient sample strategy to explore neighbourhood
- Sample paths from starting nodes
- Paths as “sentence” - equivalent to words

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{v,x}}{Z}, & \text{if } (v,x) \in E \\ 0, & \text{otherwise} \end{cases}$$

$$\pi_{v,x} = w_{v,x}; Z = |E_v|$$

Random Walk in node2vec

- 2^{nd} order Random Walk
- Return parameter p
- In-out parameter q
- BFS & DFS
  - homophily vs. structural similarity

$$\alpha_{p,q}(t,x) = \begin{cases} \frac{1}{p}, & \text{if } d_{t,x} = 0 \\ 1, & \text{if } d_{t,x} = 1 \\ \frac{1}{q}, & \text{if } d_{t,x} = 2 \end{cases}$$

$$\pi_{v,x} = \alpha_{p,q}(t,x) * w_{v,x}$$

Random Walk in node2vec

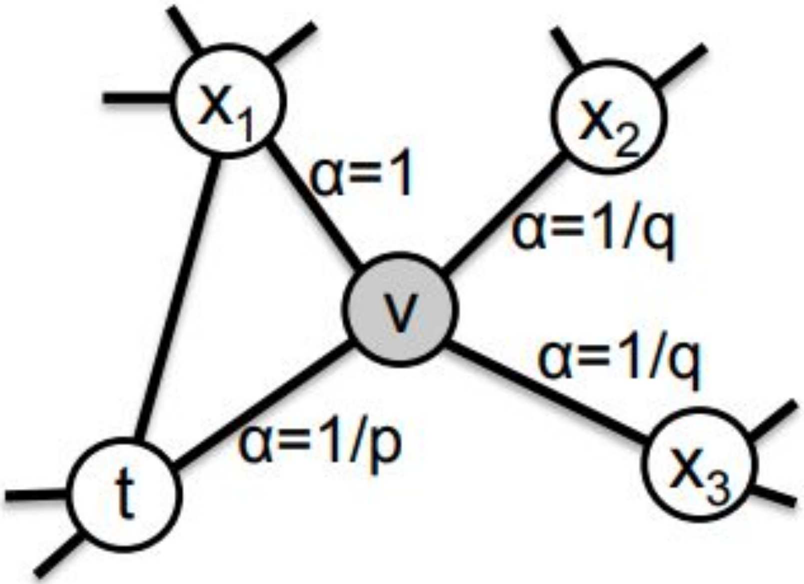


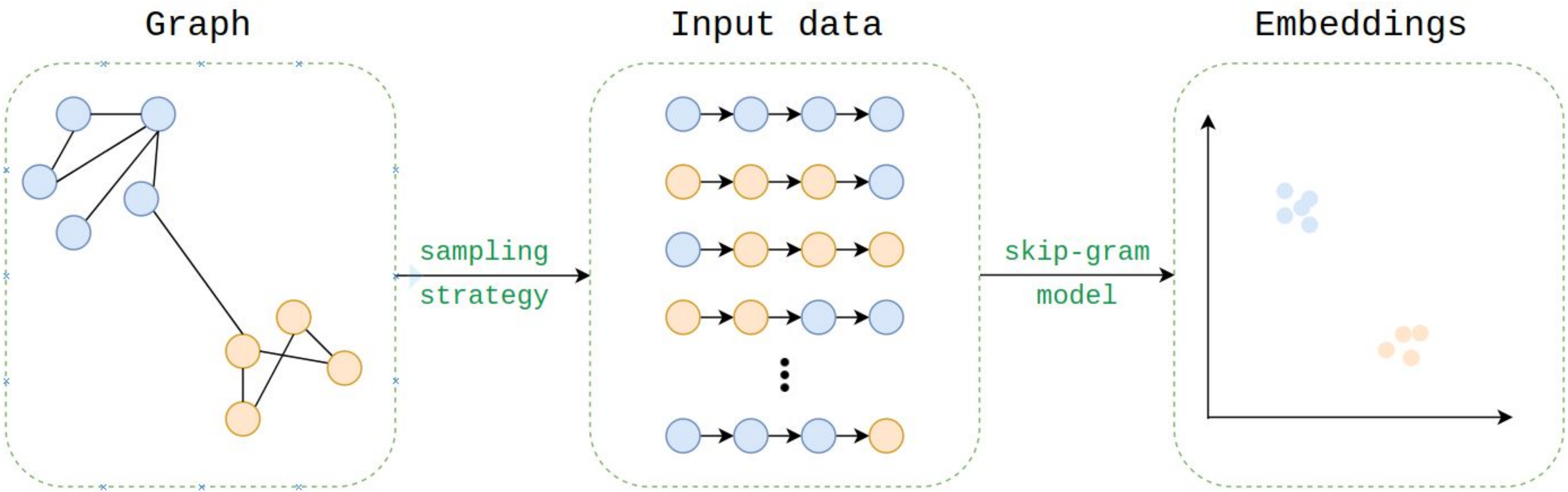
figure 3: Grover et al. 2016

$$\alpha_{p,q}(t,x) = \begin{cases} \frac{1}{p}, & \text{if } d_{t,x} = 0 \\ 1, & \text{if } d_{t,x} = 1 \\ \frac{1}{q}, & \text{if } d_{t,x} = 2 \end{cases}$$

$$\pi_{v,x} = \alpha_{p,q}(t,x) * w_{v,x}$$

node2vec II

figure 4: Cohen, 2018



SkipGram

- Paths instead of sentences
- Random Walks define neighbourhoods
- maximize the log Probabilities

$$\max_f \sum_{u \in V} log(\underbrace{P(N(u)|f(u))}_{\substack{\text{Probability of observing the} \\ \text{Neighbourhood of Node } u \\ \text{given the feature representation of } u}})$$

SkipGram II

Assumptions in node2vec

- conditional independence

$$P(N(u)|f(u)) = \prod_{n_i \in N(u)} P(n_i|f(u))$$

- feature space symmetry

$$P(n_i|f(u)) = \frac{exp(f(ni) * f(u))}{\sum_{v \in V} exp(f(v) * f(u))}$$

SkipGram II

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Final objective function

$$\max_f \sum_{u \in V} -log\left(\sum_{v \in V} exp(f(v) * f(u))\right) + \sum_{n_i \in N(u)} f(n_i) * f(u)$$

SkipGram II

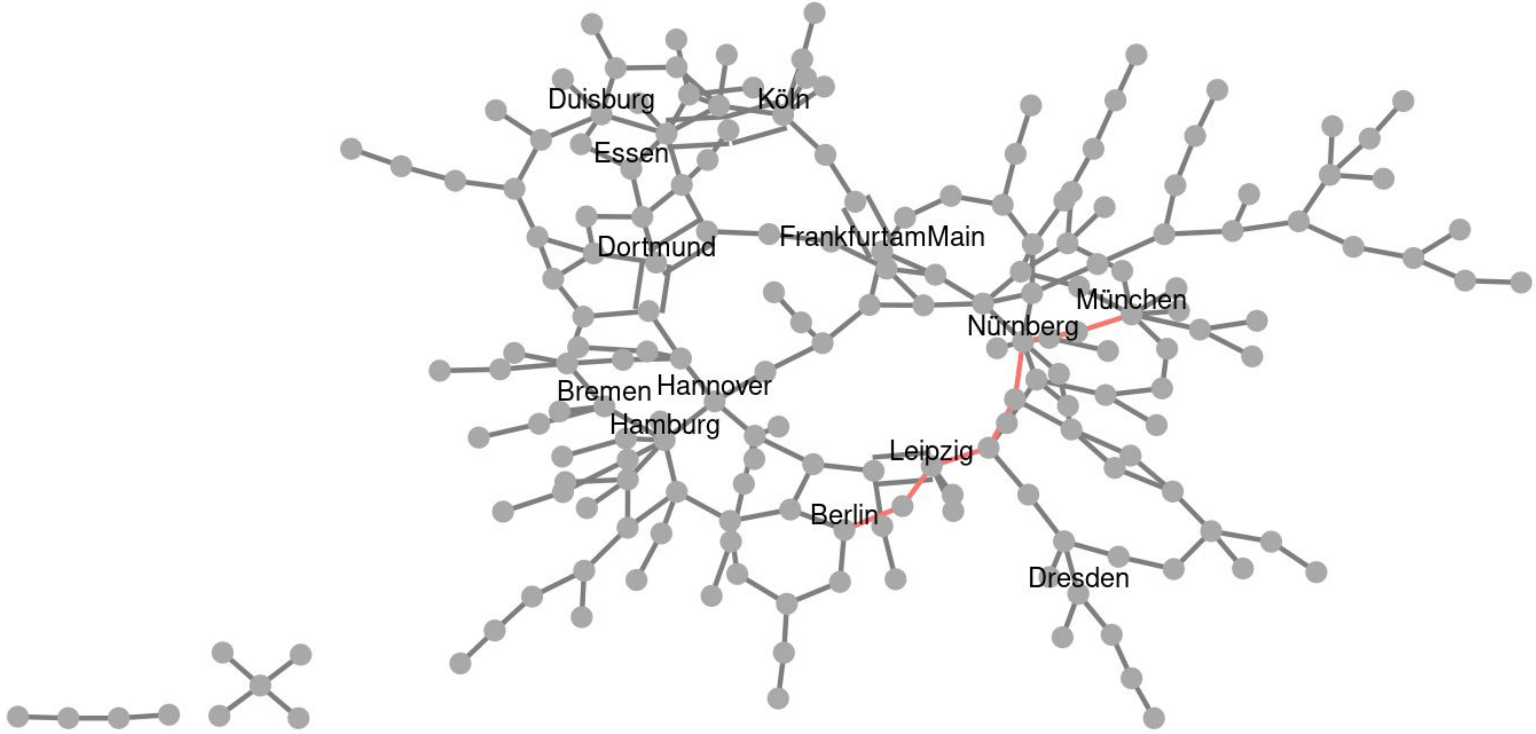
Final objective function

$$\max_f \sum_{u \in V} -log \left(\underbrace{\sum_{v \in V} exp(f(v) * f(u))}_{\text{Unfeasible to calculate}} \right) + \sum_{n_i \in N(u)} f(n_i) * f(u)$$

Negative Sampling

- Speeds up SkipGram
  - transforms probability calculation to logistic estimator
  - sampling known negative nodes
  - adjusting only relevant weights
- Alternative to hierarchical sampling (DeepWalk)

Autobahn Graph



node2vec Hyperparameters

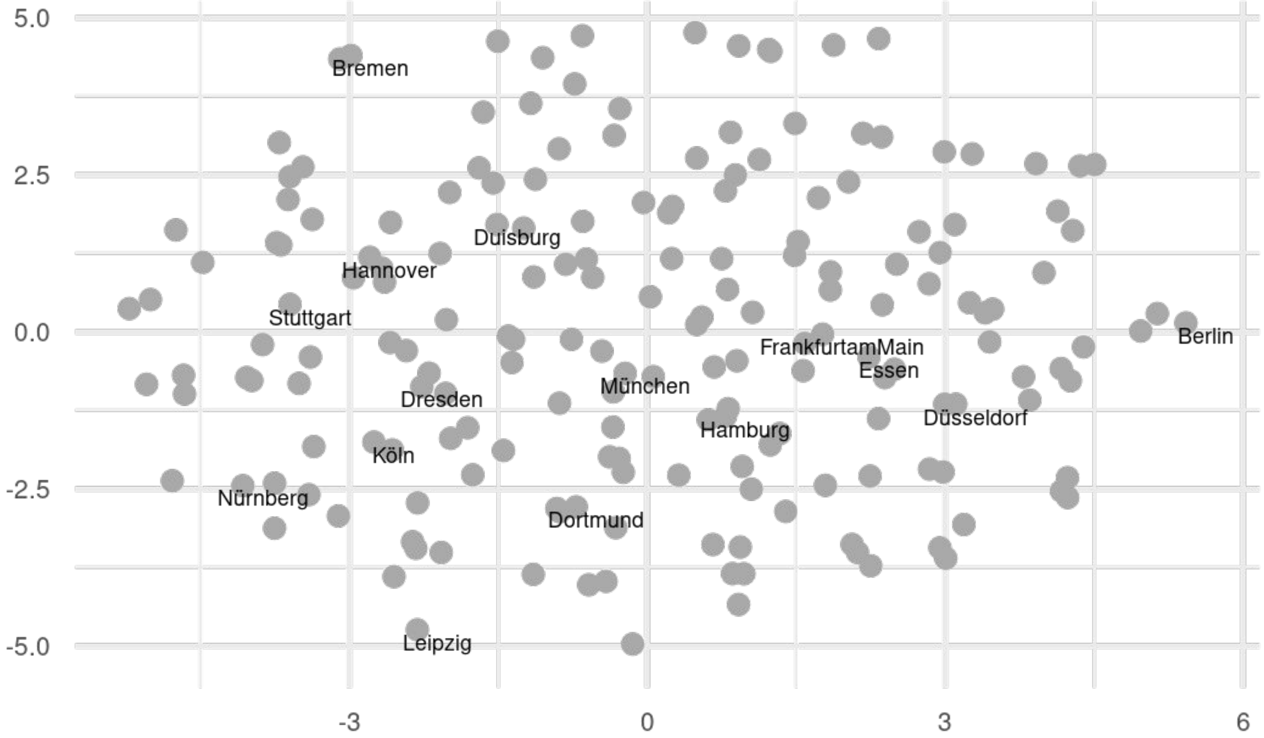
- p and q
  - usually tuned to task
  - grid search {0.25, 0.5, 0.75, 1, 2, 4}
- number of walks
- length of walks

node2vec Hyperparameters

Here:

- p = 0.5
- q = 1
- number of walks = 10
- length of walks = 12

Embedded Autobahn Example (tSNE)



Use Cases NLP

- Text evaluation/classification
- Social Networks
- Semantic searches
- Fake news
- Speech recognition
- Translation
- Chat Bot
- etc.

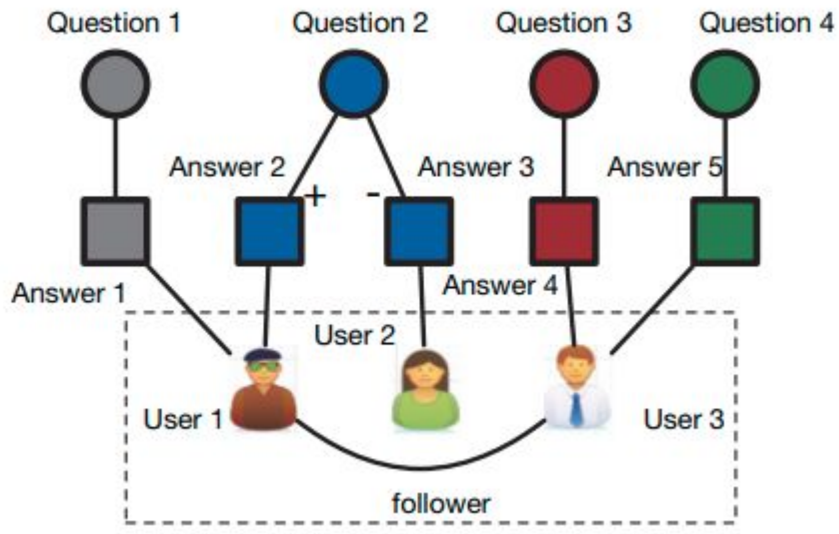
Q&A / CQA

- Community based Question & Answer Forums
  - stackoverflow, Quora, etc.
- Goals
  - Finding best Answer
  - similar Question
  - Expert on a topic

Q&A Example Dataset

- Quora heterogeneous dataset
    - 444˙138 questions
    - 95˙915 users
    - 887˙771 answers

figure 5: Fang et al. , 2016



Recurrent Neural Network (RNN)

- Previous “state” influences the current output
- Temporal dynamic behavior
- Sequence or Stream of Data
  - LSTM with Random Walks
    - common use in CQA tasks

Outlook Use Cases Graph Embedding

- Computer Vision
  - Image captioning
  - Video tracking
- Biological
  - Brain network representation
  - Genomic

References

- Karte: NordNordWest, Lizenz: Creative Commons by-sa-3.0 de, CC BY-SA 3.0 DE <https://creativecommons.org/licenses/by-sa/3.0/de/deed.en>, via Wikimedia Commons (figure 1)
- Cai, HongYun, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. “A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications.” IEEE Transactions on Knowledge and Data Engineering 30 (9): 1616–37. https://doi.org/10.1109/TKDE.2018. 2807452. (figure 2)
- Grover, Aditya, and Jure Leskovec. 2016. “Node2vec: Scalable Feature Learning for Networks.” CoRR abs/1607.00653. http://arxiv.org/abs/1607.00653. (figure 3)
- Cohen, Elior. 2018. “Node2vec: Embeddings for Graph Data.” Towards data science. https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef. (figure 4)
- Fang, H., Wu, F., Zhao, Z., Duan, X., Zhuang, Y., & Ester, M. (2016). Community-Based Question Answering via Heterogeneous Social Network Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). (figure 5)

Discussion